DESIGN OF ENERGY EFFICIENT HIGH SPEED I/O INTERFACES

BY

MRUNMAY VYANKATESH TALEGAONKAR

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Doctoral Committee:

Associate Professor Pavan Kumar Hanumolu, Chair
Associate Professor Rakesh Kumar
Professor Elyse Rosenbaum
Professor Naresh Shanbhag

ABSTRACT

Energy efficiency has become a key performance metric for wireline high speed I/O interfaces. Consequently, design of low power I/O interfaces has garnered large interest that has mostly been focused on active power reduction techniques at peak data rate. In practice, most systems exhibit a wide range of data transfer patterns. As a result, low energy per bit operation at peak data rate does not necessarily translate to overall low energy operation. Therefore, I/O interfaces that can scale their power consumption with data rate requirement are desirable. Rapid on-off I/O interfaces have a potential to scale power with data rate requirements without severely affecting either latency or the throughput of the I/O interface. In this work, we explore circuit techniques for designing rapid on-off high speed wireline I/O interfaces and digital fractional-N PLLs.

A burst-mode transmitter suitable for rapid on-off I/O interfaces is presented that achieves 6 ns turn-on time by utilizing a fast frequency settling ring oscillator in digital multiplying delay-locked loop and a rapid on-off biasing scheme for current mode output driver. Fabricated in 90 nm CMOS process, the prototype achieves 2.29 mW/Gb/s energy efficiency at peak data rate of 8 Gb/s. A 125X (8 Gb/s to 64 Mb/s) change in effective data rate results in 67X (18.29 mW to 0.27 mW) change in transmitter power consumption corresponding to only 2X (2.29 mW/Gb/s to 4.24 mW/Gb/s) degradation in energy efficiency for 32-byte long data bursts. We also present an analytical bit error rate (BER) computation technique for this transmitter under rapid on-off operation, which uses MDLL settling measurement data in conjunction with always-on transmitter measurements. This technique indicates that the BER bathtub width for 10^{-12} BER is 0.65 UI and 0.72 UI during rapid on-off operation and always-on operation, respectively.

Next, a pulse response estimation-based technique is proposed enabling burst-mode oper-

ation for baud-rate sampling receivers that operate over high loss channels. Such receivers typically employ discrete time equalization to combat inter-symbol interference. Implementation details are provided for a receiver chip, fabricated in 65nm CMOS technology, that demonstrates efficacy of the proposed technique. A low complexity pulse response estimation technique is also presented for low power receivers that do not employ discrete time equalizers.

We also present techniques for implementation of highly digital fractional-N PLL employing a phase interpolator based fractional divider to improve the quantization noise shaping properties of a 1-bit $\Delta\Sigma$ frequency-to-digital converter. Fabricated in 65nm CMOS process, the prototype calibration-free fractional-N Type-II PLL employs the proposed frequency-to-digital converter in place of a high resolution time-to-digital converter and achieves 848 $\mathrm{fs_{rms}}$ integrated jitter (1 kHz-30 MHz) and -101 dBc/Hz in-band phase noise while generating 5.054 GHz output from 31.25 MHz input.

To my parents, for their love and support.

ACKNOWLEDGMENTS

I have been very fortunate to have met so many people during my Ph.D. who made all these years very memorable. I hope that their friendship, goodwill and support will accompany me in the future as I step back into the real world.

First and foremost among them is my advisor, Prof. Pavan Kumar Hanumolu. He gave me an opportunity to work with him as well as the fantastic set of students in his group. I am grateful to him for giving me this opportunity. It has been a great honor and pleasure to work with him. He is one of the most down-to-earth people I have known. His door is always open for the students despite his busy schedule, something that has never ceased to amaze me. His patience, humility and friendliness have set a great example for me. I hope I can always count on his advice in the future as I have done so far.

I am also indebted to my advisor at IIT Madras, Prof. Shanthi Pavan, for getting me started on this path. Not only did he provide an inspiration, he also supported me as a Project Associate before I began my Ph.D. I am also thankful to Prof. Moon and Prof. Mayaram for their guidance and support. I would also like to thank Prof. Rakesh Kumar, Prof. Rosenbaum and Prof. Shanbhag for being on my doctoral committee and for providing very helpful feedback on my research. Thanks are also due to Laurie Fischer and Prof. Franke for making my transfer from Oregon State University to University of Illinois very smooth. I am thankful to Rachel Glasa for her help in navigating the administration maze. I am grateful to Srikanth Gondi and Bryan Casper for their mentorship as well as for giving me opportunities to do internships at Silicon Image and Intel Labs, respectively.

I have been lucky enough to meet friends who were not only great fun but also the ones I could lean on for help. The company of Ankur, Hari, Karthik, Manideep, Praveen, Sachin and Sameer made sure that dull times were few and rare at Oregon State University, while

Ahmed, Guanghua, Romesh, Saurabh, Seong-Joong, Tejasvi and Wooseok kept the party going after transfer to University of Illinois. As seniors from the group, Rajesh and Amr have always been forthcoming with their advice on both technical and non-technical matters. The recent additions to the group, Charlie (not his real name), Dongwook, Mostafa, Safwat and Timir, are a pleasure to work with. I hope they'll have as much if not more fun as I did during my Ph.D. Thank you everyone for your friendship!

I cannot thank my parents enough for their unconditional love, support and encouragement. I am also grateful to my sister, Vaishnavi, for her love and understanding as well as for graciously suffering through my bouts of sarcasm.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

With the advent of Web2.0 and the age of Big Data, data bandwidth requirements have seen an explosive growth. Computing systems that cater to this demand have also seen great performance improvements over the years due to progress in semiconductor manufacturing process technologies and architecture design. While initial performance gains came about due to increased clocking frequencies, more recently designers have resorted to parallelism for performance improvement. High performance computing systems such as data centers and supercomputers rely on scores of processor chips processing data in parallel. A similar trend of multi-core processing is also seen in desktop computers as well as mobile devices. As a result, the performance of computing systems is increasingly being limited by how fast the data can be transferred to and from the system, be it from a processor to memory or from a processor to another processor/peripheral device. On the other hand, energy efficiency has also become a key concern, reflected in power and cooling costs for large scale computing systems and in battery-life for small scale systems such as hand-held devices. Figure 1.1 shows the plot of aggregate I/O bandwidth of recently published high performance microprocessors. The trend predicts that by year 2020, aggregate I/O bandwidth requirement of microprocessors will exceed 1 TB/s. For such bandwidth requirements, power consumed by the I/O interfaces may account for 50% of the processor thermal design power [1]. A similar increase in I/O bandwidth can also be expected for peripheral devices such as storage and network interfaces. The need for low power I/O interfaces is equally acute in power-constrained mobile computing platforms. Therefore, design of energy efficient I/O interfaces is important to sustain the growth in performance of modern computing systems.

In this work, we explore system-level techniques for the design of energy efficient high

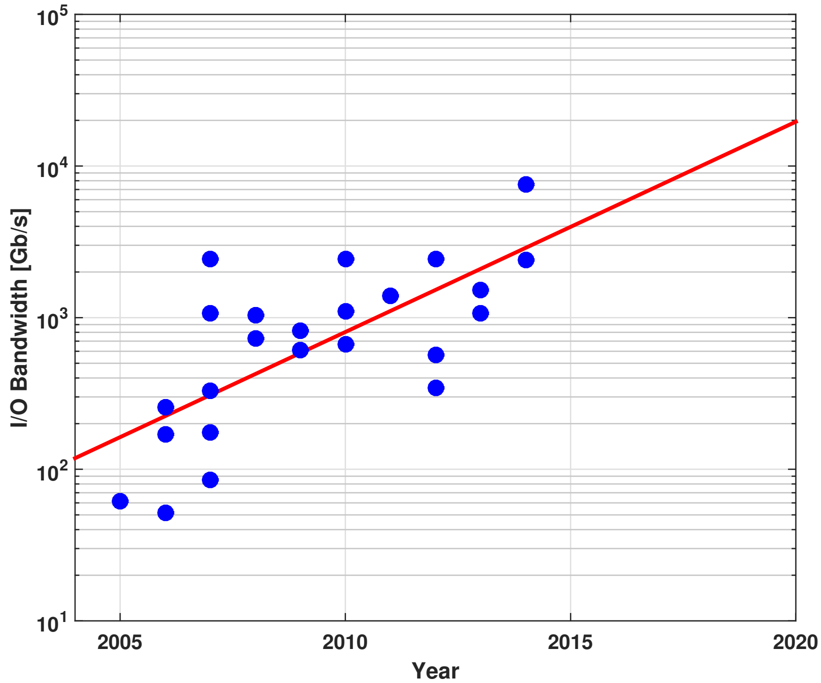


Figure 1.1: Aggregate I/O bandwidth of recently published high performance microprocessors.

speed I/O interfaces. These techniques exploit the varying data transfer bandwidth requirements at system level to save I/O interface power. To explain these system-level techniques and proposed circuit implementations, this dissertation is organized as follows. Chapter 1 provides a brief overview of high speed wireline I/O interfaces used for chip-to-chip communication. Chapter 2 describes system-level energy efficiency techniques and energy efficiency metrics. Implementation details of a rapid on-off transmitter chip are given in Chapter 3 and those of a burst-mode receiver are given in Chapter 4. A technique for estimation of pulse response of a channel is described in Chapter 5. Chapter 6 discusses design of a high digital fractional-N PLL and Chapter 7 provides the conclusion.

In this chapter we begin with a short overview of evolution of high speed I/O interfaces in Section 1.1, followed by a description of I/O interface architectures in Section 1.2. I/O interface performance metrics and active power reduction techniques are described in Section 1.3, and Section 1.4 respectively.

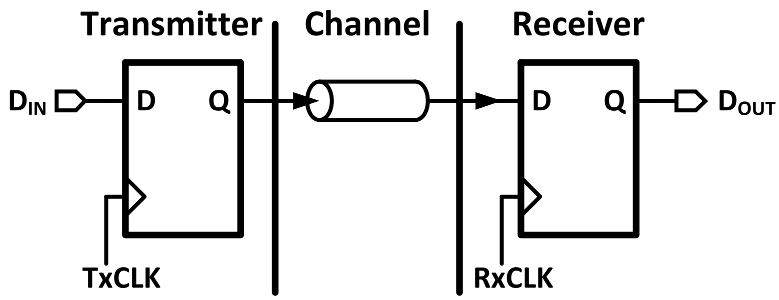


Figure 1.2: Basic I/O interface.

1.1 Evolution of High Speed I/O Interfaces

In its most basic form, shown in Fig. 1.2, a chip-to-chip I/O interface consists of a flip-flop on both transmitter and receiver sides. The interconnection between transmitter and receiver is made using a channel such as a printed circuit board (PCB) trace or a copper wire. The flip-flops on the transmitter and the receiver are clocked by TxCLK and RxCLK, respectively. Suppose TxCLK and RxCLK have nominally the same frequency and the difference between their positive edge time instants is ΔT. The timing skew, ΔT, between TxCLK and RxCLK is positive when TxCLK is delayed with respect to RxCLK. ΔT is also assumed to be bounded between $\pm 0.5\mathrm{T_B}$. To ensure that the receiver samples the data correctly, the following condition must be satisfied:

$$\mathrm{T_B} \quad > \quad \Delta\mathrm{T} + \mathrm{T_{Tx,C\text{-}Q}} + \mathrm{T_{Rx,SU}} + \mathrm{T_{ch}} - \left(\left\lfloor \frac{\mathrm{T_{ch}}}{\mathrm{T_B}} \right\rfloor \right) \mathrm{T_B} \tag{1.1}$$

where $\mathrm{T_B}$ is the period of RxCLK as well as bit period, $\mathrm{T_{Tx,C\text{-}Q}}$ is the clock to Q delay of transmitter side flip-flop, $\mathrm{T_{Rx,SU}}$ is the setup time of the receiver side flip-flop, and $\mathrm{T_{ch}}$ is the time taken by the signal to travel from transmitter output to receiver input.

At multi-Gb/s data rate, the timing uncertainty ΔT becomes a significant fraction of bit period $\mathrm{T_B}$. The distributed nature of the channel also comes into view at high data rates, causing signal attenuation as well as reflections. This necessitates using an appropriately terminated output driver on the transmitter. On the receiver side, a termination as well as sampling block that compares input signal with a reference voltage $\mathrm{V_{REF}}$ is required, as shown in Fig. 1.3. A deskew block is used to cancel the effect of static timing uncertainty

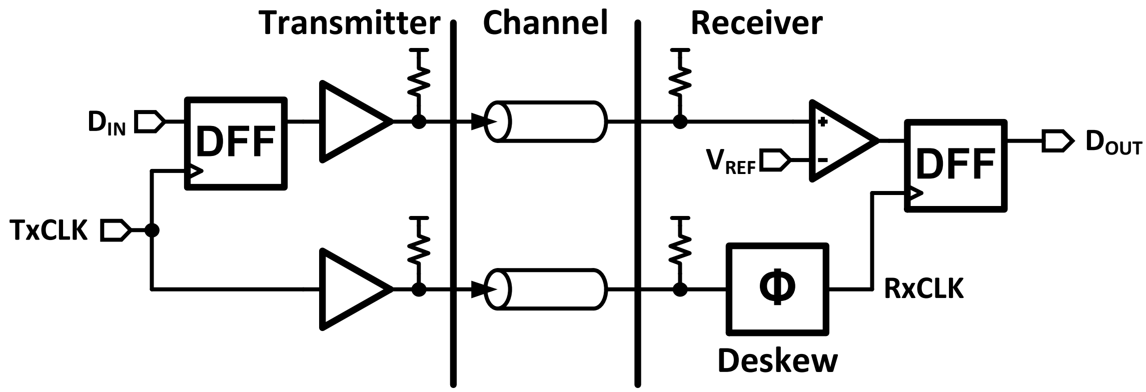


Figure 1.3: I/O interface with clock deskew.

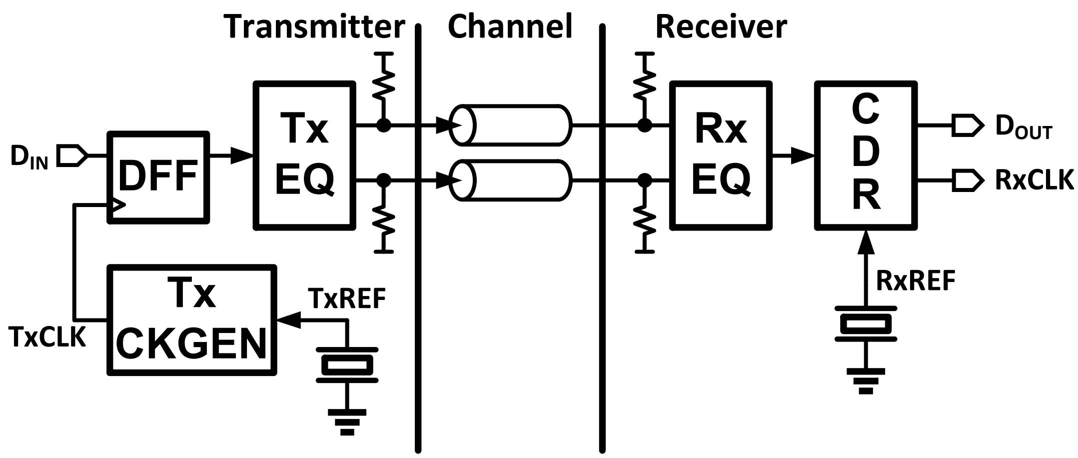


Figure 1.4: I/O interface with equalization, clock generation and clock recovery.

ΔT as well as channel delay T_{ch} on sampling time instant.

At even higher data rate, dynamic changes in timing uncertainty, ΔT, due to noise, crosstalk, and environmental conditions cannot be ignored. Such dynamic timing uncertainties are also called jitter. Figure 1.4 illustrates an I/O interface that uses a clock and data recovery (CDR) block to track static as well as dynamic timing uncertainties. It also shows that in practice, a clock multiplier (Tx CKGEN in Fig. 1.4) multiplies a low frequency crystal reference clock frequency to generate the high frequency clock TxCLK. Similarly, a low frequency reference clock RxREF may also be utilized by the CDR block. To combat channel loss at high data rate, differential signaling as well as transmitter equalizer (TxEQ) and receiver equalizer (RxEQ) are used.

1.2 High Speed I/O Interface Architectures

While many I/O interface architectures such as multi-drop bus-based [2], simultaneous bidirectional [3], single ended signaling-based [4] exist, we limit our discussion to point-to-point unidirectional differential signaling-based I/O interfaces. Such interfaces are more robust to channel non-idealities as well as crosstalk and therefore are commonly used at high data rates. There are two main types of point-to-point I/O interface architectures:

- Forwarded clock architecture
- Embedded clock architecture

Details of these architectures are as follows.

1.2.1 Forwarded Clock Architecture

For I/O interfaces with many parallel lanes, an efficient way of tracking jitter is to send clock signal from transmitter to the receiver. Forwarded clock architecture uses this principle to improve jitter tracking. A simplified block diagram of an I/O interface with forwarded clock architecture is shown in Fig. 1.5. For simplicity, a differential signaling lane is shown as a single lane in this figure. On the transmitter side, data is retimed in the RET block using transmitter clock, TxCLK. In addition to data, clock is also sent to the receiver, shown as FWDCLK in Fig. 1.5. On the receiver side, this clock is distributed to all the data lanes. The receiver samples the data using samplers denoted as SAMP. CDR block along with De-skew block varies the delay of RxCLK to provide the clock for the samplers. The HyperTransport I/O interface [5] is an example of forwarded clock architecture.

1.2.2 Embedded Clock Architecture

For I/O interfaces with fewer or longer lanes, power overhead of sending clock from transmitter to the receiver may become unacceptable. In such cases the receiver extracts the clock information from the data input using the CDR block. A simplified block diagram of I/O interface with embedded clock architecture is shown in Fig. 1.6. Unlike the transmitter in

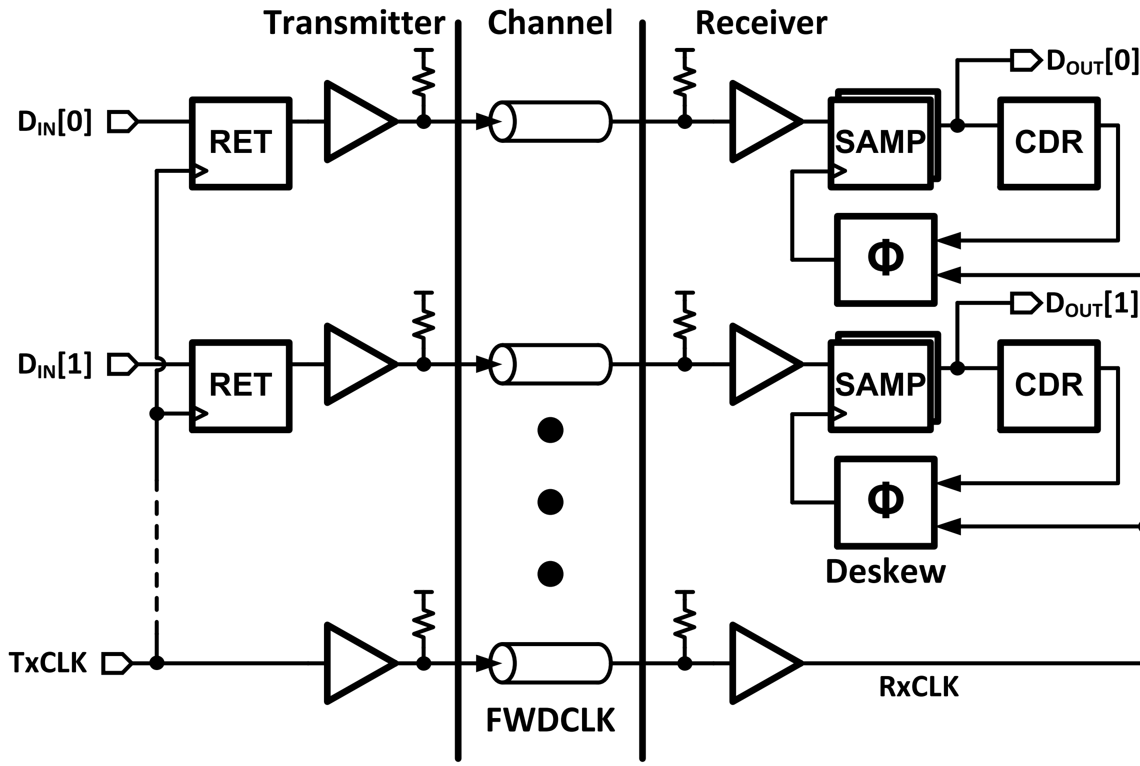


Figure 1.5: Simplified block diagram of I/O interface with forwarded clock architecture.

forwarded clock architecture, the transmitter in embedded clock architecture does not have an extra clock forwarding lane. At the receiver, the input data is sampled using output of a clock generator block (CLKGEN) that is controlled by the CDR block. A voltage controlled oscillator (VCO) is often used as clock generator in embedded clock architecture interfaces. The PCI Express I/O interface [6] is an example of embedded clock architecture.

1.3 I/O Interface Performance Metrics

Timing uncertainty, also called jitter, plays an important role in determining many performance metrics of a high speed I/O interface. Jitter can be defined as deviation of data or clock edge from ideal timing [7]. There are two main sources of jitter in an I/O interface. First, random noise caused by circuit elements as well as limited bandwidth of circuits result in jitter. Channel is the second main source of jitter due to its bandwidth limitations. With this in mind, we next consider the performance metrics associated with I/O interfaces.

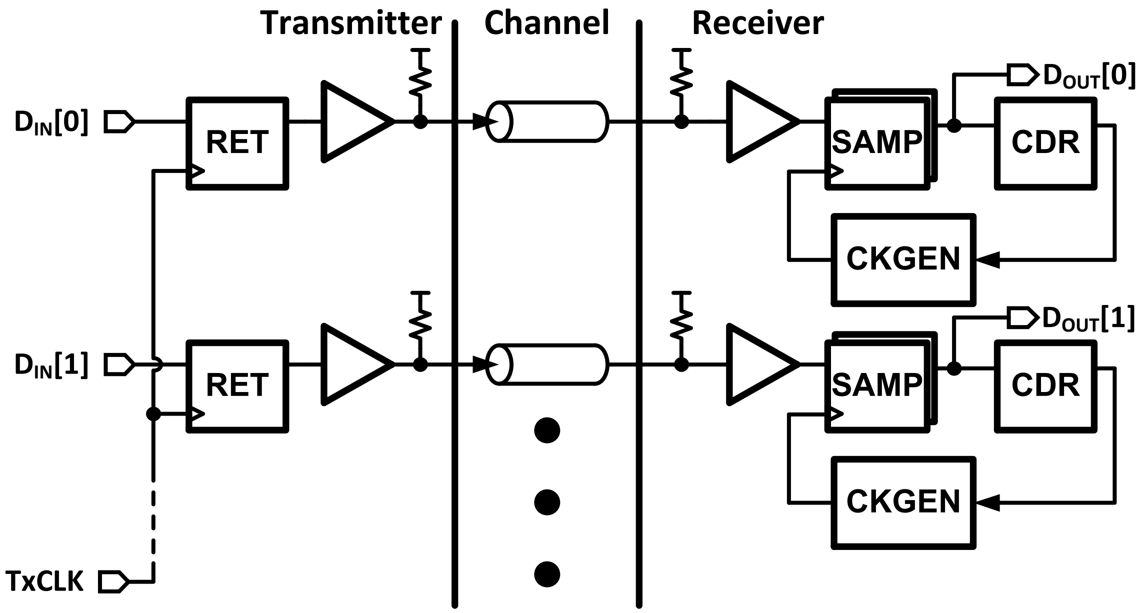


Figure 1.6: Simplified block diagram of I/O interface with embedded clock architecture.

1.3.1 Eye Diagram

To ensure transmitter performance and channel compliance an eye mask is specified at the receiver input. The eye mask delineates the minimum area for the eye opening. An example of transmitter eye mask is shown in Fig. 1.7. A larger eye opening implies better transmitter performance.

1.3.2 Rx Bit Error Rate

There are two main methods of specifying the receiver (or I/O interface) bit error rate (BER) performance as explained below:

BER Bathtub:

The BER bathtub is commonly used to characterize the BER performance of forwarded clock receivers and I/O interfaces. It is a plot of BER as function of phase offset from the center of the data eye. An example of BER bathtub characteristic is shown in Fig. 1.8. A wider BER bathtub characteristic at a given BER is preferred.

Jitter Tolerance:

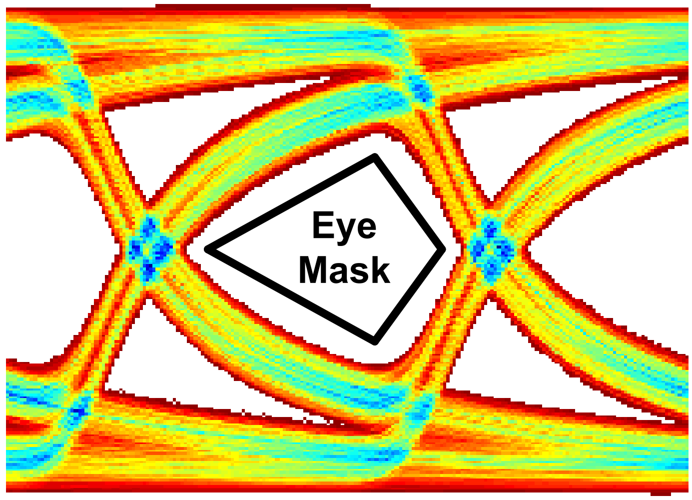


Figure 1.7: An example of eye diagram with eye mask.

Jitter tolerance (JTOL) specifies the maximum amplitude of input sinusoidal jitter that a CDR can tolerate to achieve a specific BER. JTOL is commonly used to characterize embedded clock receivers. Figure 1.9 shows the JTOL mask specified for PCI express 3.0 [8].

1.3.3 Energy Efficiency

Energy efficiency of an I/O interface is the average energy spent per bit by the interface for data transfer. It is also the ratio of power consumed by an I/O interface and its data rate. It is expressed in pJ/b or mW/Gb/s. We will discuss this in more detail in Chapter 2.

Next, we look at various power reduction techniques used in I/O interfaces that improve their energy efficiency at peak data rate.

1.4 Active Power Reduction Techniques

The power consumed in high speed I/O interfaces can roughly be divided in two parts, namely clocking power and signaling power. Clocking power consists of power consumed in clock generation, transmitter re-timing as well as receiver CDR. On the other hand, signaling power consists of power consumed by transmitter output drivers, transmitter and receiver

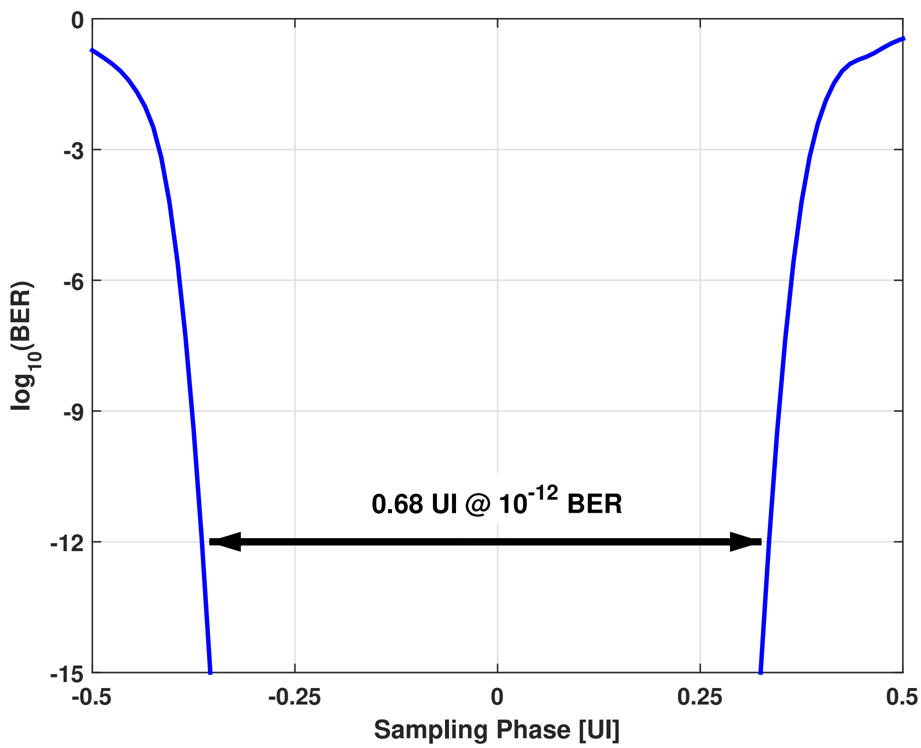


Figure 1.8: An example BER bathtub characteristic.

equalizers as well as samplers.

Choices of process technology, data rate and channel characteristics play a crucial role in determining the energy efficiency of an I/O interface. It has been suggested that for a given technology, data rate with $1\,\mathrm{UI} = 4 \times \mathrm{FO4}$ delay is optimal for balancing clocking power with data rate [9]. FO4 denotes the delay of a minimum size inverter that drives a load consisting of four identical inverters in a process technology. Sub-rate transmitter and receiver architectures that make use of lower clock frequency and time interleaving are commonly used in high speed I/O interfaces to reduce clocking power (e.g. [10]). Most I/O interfaces also share a single low jitter clock multiplier among multiple transmitter and receiver lanes to amortize its power [10]. To further reduce clocking power, deskew circuitry may also be shared among multiple receiver lanes in a forwarded clock I/O interface [11].

Low loss channels are also important for low power operation. A more lossy channel incurs higher power penalty due to increased signaling power consumed in equalizers and transmitter output drivers. For the state-of-the-art I/O interfaces with high loss channels with around

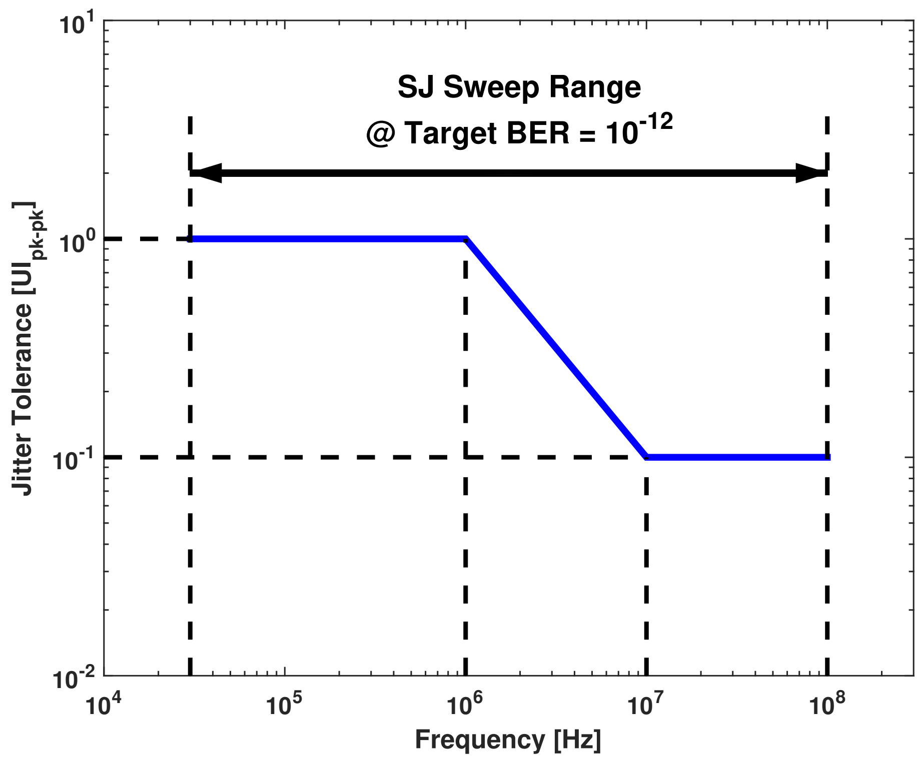


Figure 1.9: PCI Express 3.0 jitter tolerance mask.

30 dB loss at Nyquist frequency, the power efficiency is in the order of 20 pJ/bit [12]. On the other hand, for low loss channel I/O interfaces with $<$ 15 dB loss at Nyquist frequency, the power efficiency is typically less than 5 pJ/bit [1].

A highly sensitive receiver improves the active power efficiency by a huge margin. As described in [13], a sensitive receiver reduces transmitter swing requirements. This not only reduces the transmitter signaling power, but also results in less loading on transmitter clock path. Therefore a sensitive receiver can reduce both signaling power and clocking power of the interface. Offset correction techniques can improve receiver sensitivity down to 8 mV_{ppd} for data rate as high as 6.25 Gb/s [10]. In case of multi-lane parallel I/O interfaces, distribution of a high frequency clock can also consume considerable power. Clock distribution techniques such as resonant clock distribution [10] and injection locked clock distribution [14] can result in significant power savings. Further savings in clock distribution power can be achieved by distributing a low frequency clock which is then locally multiplied for each transceiver lane [15].

Joint optimization of the transmitter as well as receiver power is also important in achieving better energy efficiency. Receiver power consumption increases with improved input sensitivity. On the other hand, transmitter power reduces with lower output swing requirement. Therefore, there exists a point where transmitter and receiver power consumption is balanced resulting in the lowest energy efficiency for the I/O interface [16].

The emphasis of these techniques is on improving the energy efficiency of the high speed I/O interfaces at the peak performance. In practice most of the systems under-utilize their performance capability. Therefore it is beneficial to make these systems more efficient for the real world usage patterns. For example, many I/O interfaces cater to burst mode data traffic, as opposed to uniform data traffic. So I/O interfaces that maintain their energy efficiency even under burst mode data traffic would be desirable. By exploiting data transfer patterns, system level techniques such as dynamic voltage and frequency scaling (DVFS) and rapid on-off (ROO) can save substantial amount of power in I/O interfaces. These system level techniques and their impact on energy efficiency are studied in the next chapter.

CHAPTER 2

SYSTEM-LEVEL TECHNIQUES FOR ENERGY EFFICIENCY

The key to system level energy efficiency improvement techniques is the observation that most of the computing systems and networks rarely operate at their peak utilization levels [17, 18]. Figure 2.1 shows the average utilization profile for server central processing units (CPUs). It can be seen that most of the time CPUs operate in 10%-50% utilization region. It would be reasonable to assume that the I/O utilization profile will also be similar. But for a fixed power I/O interface the energy per bit increases as average data rate reduces. This is because the interface consumes the same amount of power to transfer data at lower average data rate. To maintain energy efficiency of the I/O interface, its power should also scale with average data rate.

Dynamic voltage and frequency scaling (DVFS) and rapid on-off (ROO) are two main techniques used to design energy efficient high speed I/O interfaces that consume power in proportion to their performance, i.e. average data rate. In I/O interfaces utilizing DVFS, the supply voltage and data rate are scaled based on the data rate requirement. For example, supply voltage and data rate can be reduced when the link is handling less data. Similarly in ROO I/O interfaces, the interface transfers the data at its peak data rate and turns off as soon as transfer of the data is over. In this chapter we provide details of two system-level energy efficiency techniques, namely DVFS and ROO, as well as evaluate their effectiveness based on various energy efficiency metrics. After briefly describing prior art in DVFS and ROO I/O interface design in Section 2.1, we discuss their implementation issues in Section 2.2. System level energy efficiency metrics are defined in Section 2.3 and are evaluated for DVFS and ROO I/O interfaces using queue-based models in Section 2.4.

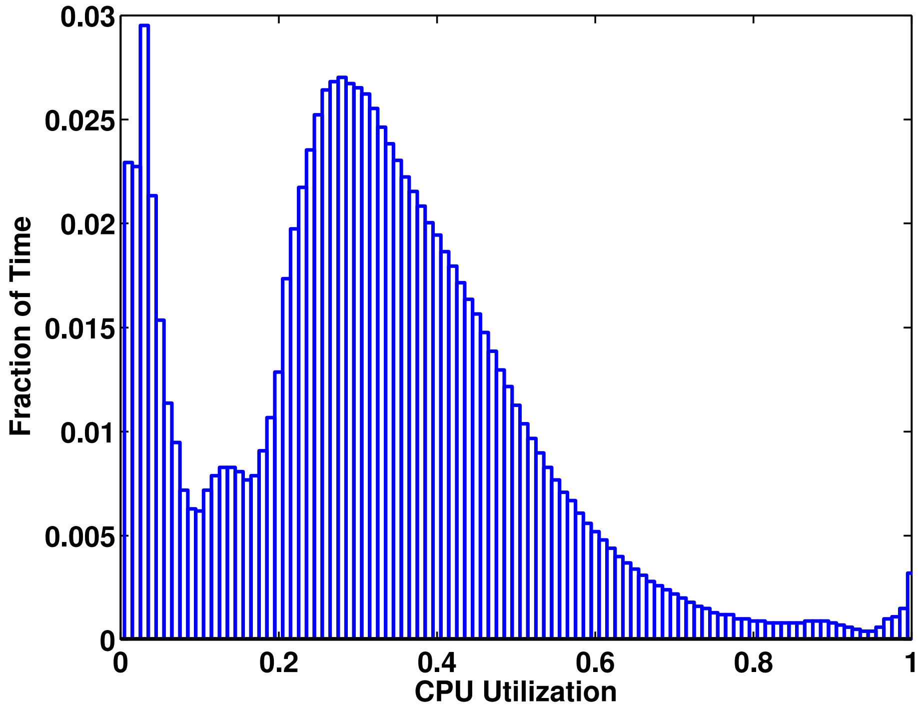


Figure 2.1: Average utilization profile of server CPUs [17].

2.1 Prior Art

2.1.1 DVFS I/O Interfaces

Following the use of DVFS in microprocessors and other digital systems on chip (SoCs), a frequency scalable parallel I/O interface was proposed in [19]. In this implementation, a ring oscillator is used to come up with an appropriate supply voltage for a given data rate decided by input reference frequency F_{REF}. The half-duplex forwarded clock parallel I/O interface uses dual loop delay-locked loop (DLL) architecture for data recovery. The chip is fabricated in 0.35 μm CMOS technology with maximum supply voltage of 3.3 V. The supply voltage range for the interface is from 1.3 V to 3.2 V while operating at a variable data rate from 0.2 Gb/s to 1 Gb/s. The lower limit on supply voltage is due to the receiver samplers. The DC-DC converter is also on-chip except for the filter inductor and capacitor.

The adaptive supply embedded clock serial link proposed in [20] can operate at a variable data rate from 0.65 Gb/s to 5 Gb/s. This chip is fabricated in a 0.25 μm CMOS process with

Table 2.1: Power states in [22].

State	Description	Exit Lat. @4.3 Gb/s	Power (Norm. to P_{active})
P3	Synchronous Pause	0	15.2%
P2	Power down Tx/Rx front-end, regulator, bias	18.6 ns	5.7%
P1	Power down CMU	241.9 ns	0.35%

maximum supply voltage of 2.5 V. The link supply voltage can vary from 0.9 V to 2.5 V. The DC-DC converter operates with 83%-94% efficiency. The settling time of the DC-DC converter for 99% accuracy is reported to be 80 μs.

A scalable I/O interface is also demonstrated in [21]. But this implementation does not have an on-chip adaptive supply generator. The I/O interface can operate at a data rate from 5 Gb/s to 15 Gb/s with supply voltage varying from 1.05 V to 0.68 V. The chip is fabricated in a 65 nm CMOS process.

2.1.2 ROO I/O Interfaces

The potential of fast transition low power states to improve energy efficiency of an I/O interface is utilized in [22]. A bidirectional asymmetric I/O interface is proposed which does not require a PLL or a DLL on the slave side. Such asymmetric interfaces are typically suitable for processor-memory interfaces. This design incorporates multiple power states which are summarized in Table 2.1. Going from power state P3 to P1, progressively more circuits are turned off for lower power at the cost of increased exit latency. This design is fabricated in a 40 nm CMOS process and each link can operate at 4.3 Gb/s data rate with 1.1 V supply voltage.

In [23] only two modes are made available for the I/O interface. If the lowest power mode has very low exit latency, all other power modes can be removed. This design achieves very low exit latency by using a multiplying injection locked oscillator (MILO) for clock generation. It eliminates the need for a slow clock multiplying phase-locked loop (PLL). With the use of MILO, transition time from lowest power state to active state is brought down to 8 ns. This bidirectional asymmetric interface is fabricated in a 40 nm CMOS process and can operate at 5.6 Gb/s data rate. Use of staggered turn-on for bias circuits to reduce power supply fluctuations and use of current mode logic circuits to reduce power supply induced jitter are also notable in this design.

A fast wake-up receiver operating at 10 Gb/s which returns to active state within 5 ns for a forwarded clock link is demonstrated in [11]. A combination of analog-to-digital converter and digital-to-analog converter is used to save and restore the state of receiver phase rotator.

So far it has been assumed that the required data rate is known *a priori*. In practice an I/O controller is used to schedule data transfer requests. I/O controller techniques such as buffering, scheduling, re-routing and dynamic bus width change can be used to improve I/O interface performance. For example [24] shows that it is possible to save up to 45% power in an interconnect network with 3.5% increase in latency even for moderately large exit latencies. Energy aware I/O scheduling for memory access and in multiprocessor systems is also an active area of research [24, 25, 26]. Discussion of such I/O controller techniques is out of scope of this work.

2.2 DVFS and ROO Implementation Issues

A detailed discussion of trade-offs involved in DVFS processor systems is given in [27]. Most of the issues described therein are also applicable to I/O interfaces. In addition to requiring extra area, DC-DC converters used to generate variable supply voltage have longer settling time. Due to large transition time, the DVFS I/O interfaces cannot track fast data rate requirement changes and lose power saving opportunities. The reduction in transition time is accompanied by increased ripple on the supply voltage. This may result in increased power supply induced jitter. DVFS I/O interfaces also need to wait for phase locking as and when supply voltage changes.

As process technology scales down, the supply voltage values are getting closer to threshold voltage values. Because of this, the possible supply voltage dynamic range and in turn the frequency dynamic range for DVFS keeps reducing. Increased device variability also makes it harder to design circuits which can operate reliably over a wide supply voltage range.

For all the shortcomings mentioned before, the biggest advantage of using DVFS is its potential to achieve super-linear power savings with reduced data rate as power roughly scales with cube of supply voltage value.

On the other hand, ROO I/O interface can provide power savings with reduction in data rate that scale linearly at best. Very low exit latency is critical for extracting power savings from ROO I/O interfaces. As exit latency increases, the interface cannot be turned off for a shorter amount of time resulting in loss of efficiency. To achieve short exit latencies, the jitter performance of the clock multiplier is sacrificed. Therefore, the receiver needs to have better jitter tolerance to avoid any detriment to performance.

Due to lack of complex power modes or additional supply voltage generators, the design of ROO I/O interfaces can be optimized for its performance at a single data rate. In contrast to DVFS I/O interfaces, a large decoupling capacitance on voltage supply improves the performance of the ROO I/O interface by providing the power-on surge current with little change in supply voltage.

We now define and evaluate system level energy efficiency metrics in the context of high speed I/O interfaces.

2.3 Energy Efficiency Metrics

Most of the complex electronic systems can be thought to operate in one of the following modes [28]:

- Fixed throughput mode (FTM)
- Maximum throughput mode (MTM)
- Burst throughput mode (BTM)

Energy efficiency metrics for each of the above modes of operation are discussed in [28]. These will be briefly described below in the context of I/O interfaces.

2.3.1 Fixed Throughput Mode

The systems operating in fixed throughput mode (FTM) cannot utilize any excess throughput available. Digital signal processing systems with fixed input and output rate are examples of systems operating in FTM. Energy efficiency for such systems can be defined as

$$\eta_{\mathrm{FTM}} = \frac{\mathrm{P_{total}}}{\mathrm{Throughput_{fixed}}} = \frac{\mathrm{P_{active}} + \mathrm{P_{idle}}}{\mathrm{Throughput_{fixed}}} \tag{2.1}$$

It corresponds to the energy per bit metric used for I/O interfaces. Figure 2.2 shows the plot of this metric for various types of I/O interfaces. For DVFS, the following assumptions are used along with α power law model for MOSFET [29]:

$$\begin{aligned}
\mathrm{f_{clk}} &\propto \frac{(\mathrm{V_{dd}} - \mathrm{V_{th}})^{\alpha}}{\mathrm{V_{dd}}} \\
\mathrm{P_d} &\propto \mathrm{V_{dd}^2 f_{clk}} \\
\alpha &= 1 \\
\mathrm{V_{dd,max}} &= 1\,\mathrm{V} \\
\mathrm{V_{th}} &= 0.3\,\mathrm{V}
\end{aligned}$$

2.3.2 Maximum Throughput Mode

As opposed to the systems operating in FTM, systems operating in maximum throughput mode (MTM) always require maximum possible throughput. Multi-user workstations and main-frame computers are examples of systems operating in MTM. Energy efficiency for such systems can be defined as energy-to-throughput ratio (ETR) given by

$$\eta_{\mathrm{MTM}} = \frac{\mathrm{E_{active}}}{\mathrm{Throughput_{max}}} = \frac{\mathrm{P_{active}}}{(\mathrm{Throughput_{max}})^2} \tag{2.2}$$

This metric is also equivalent to the energy-delay product metric used to measure energy efficiency of digital circuits [20]. Note that η_{MTM} does not depend on average throughput of

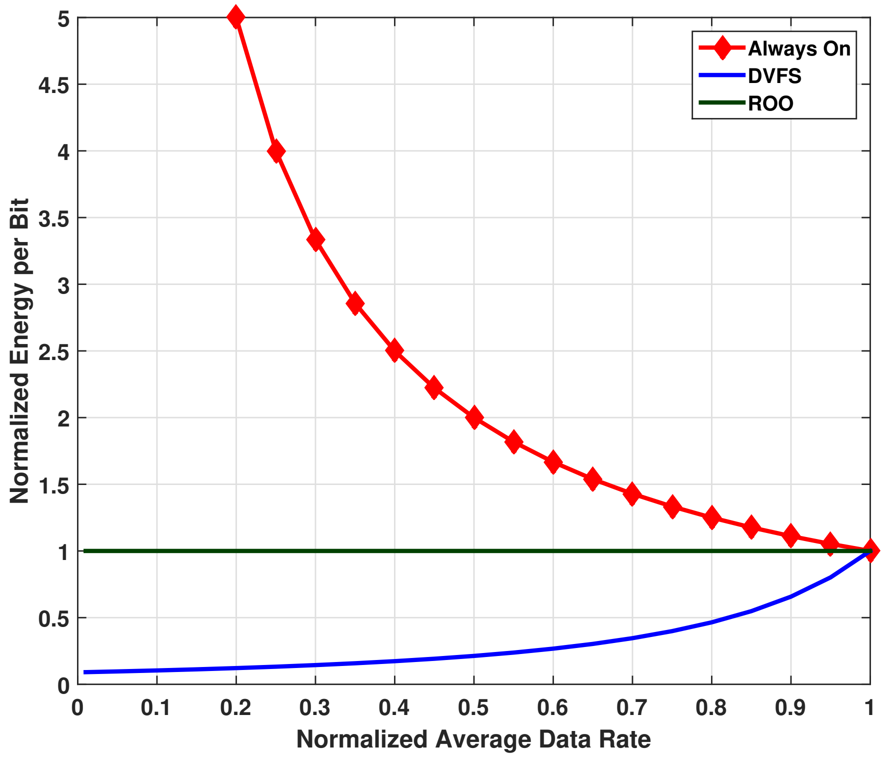


Figure 2.2: Energy efficiency for fixed throughput mode.

the system as system performance is optimized only for maximum throughput. Figure 2.3 shows the plot of this metric for various types of I/O interfaces. For DVFS the assumptions are the same as those used for FTM energy efficiency metric.

2.3.3 Burst Throughput Mode

The systems operating in burst throughput mode (BTM) are not continuously utilized. At the same time these systems should also respond as fast as possible whenever a response is expected from them. Most single user systems such as mobile devices and desktop computers operate in burst mode. Energy efficiency for such systems can be defined as burst energy-to-throughput ratio (BETR) given by

$$\begin{aligned}\eta_{\mathrm{BTM}} &= \frac{\mathrm{E_{active}} + \mathrm{E_{idle}}}{\mathrm{Throughput_{max}}} \\ &= \frac{\mathrm{P_{active}} + \mathrm{P_{idle}}}{\mathrm{Throughput_{avg}}} \times \frac{1}{\mathrm{Throughput_{max}}}\end{aligned} \tag{2.3}$$

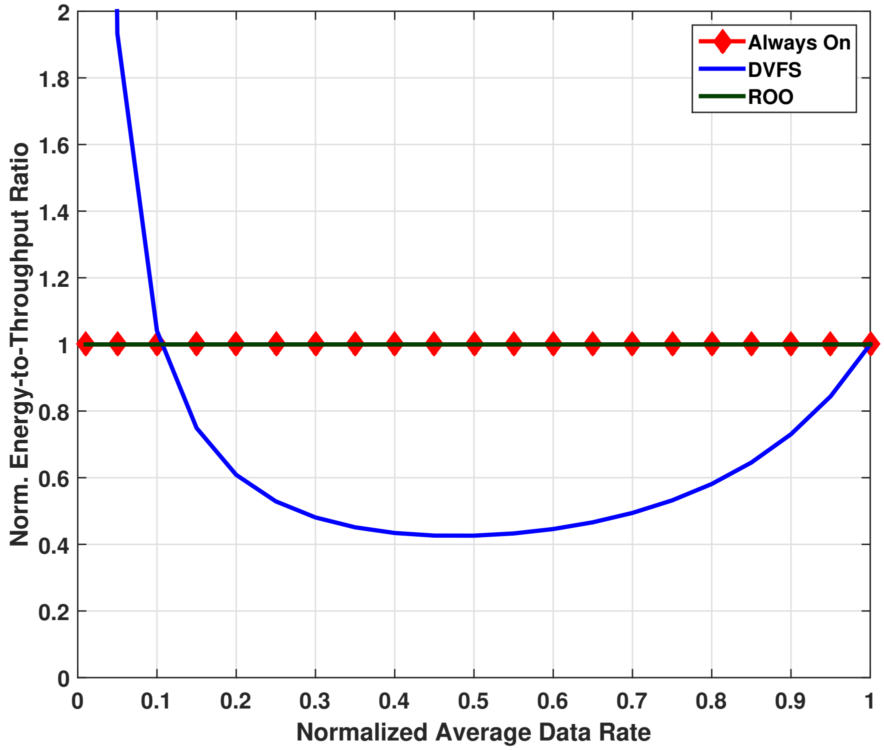


Figure 2.3: Energy efficiency for maximum throughput mode.

If β is the fraction of time spent idling and $\mathrm{P_{idle}}$ is power dissipated in idle mode, then the efficiency equation can be rewritten as

$$\eta_{\mathrm{BTM}} = \eta_{\mathrm{MTM}} + \frac{\beta \mathrm{P_{idle}}}{(\mathrm{Throughput_{max}})^2} \tag{2.4}$$

For systems operating in BTM, very low energy should be spent while idling. Simultaneously such system should also have very good energy efficiency in MTM. Figure 2.4 shows the plot of this metric for various types of I/O interfaces. These conditions are indeed reflected by the metric.

The analysis so far assumes that the data arrives at constant rate and with a fixed pattern as illustrated in Fig. 2.5. It can be seen that DVFS interface has better energy efficiency metrics in most such cases for all the three operating modes. In practice, the data transfer requests to the I/O interface may arrive at random time intervals. Energy efficiency of the I/O interface in such cases can be modeled using queues as described next.

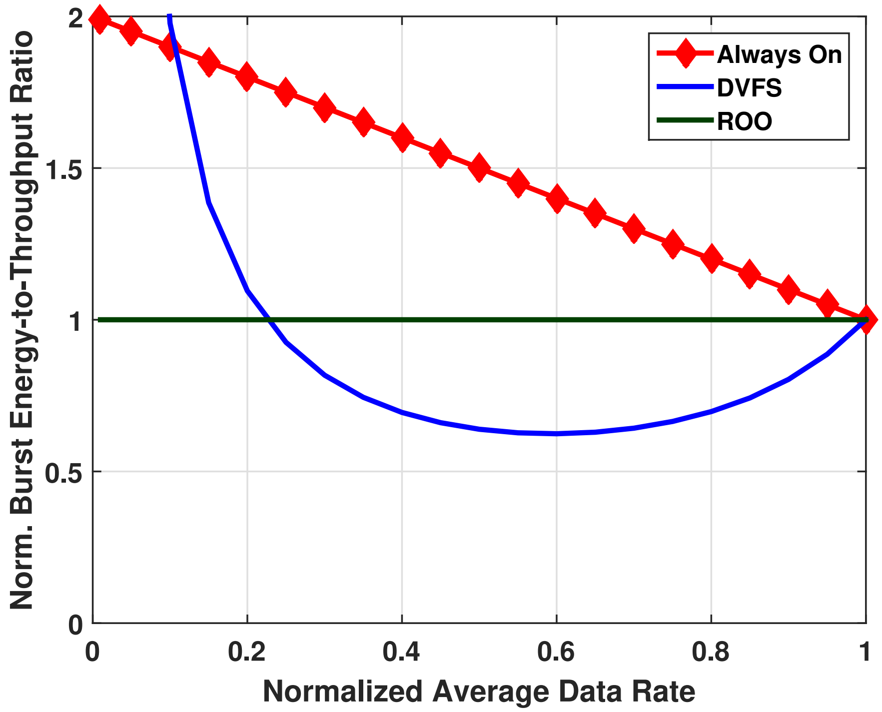


Figure 2.4: Energy efficiency for burst throughput mode.

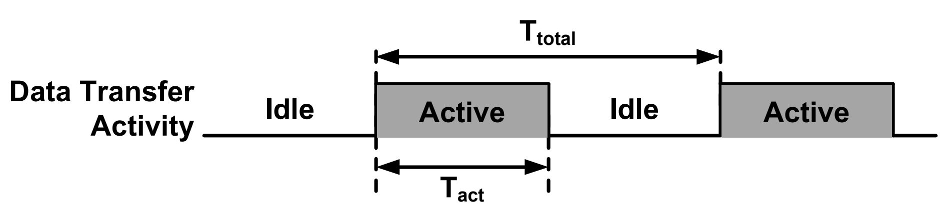


Figure 2.5: Fixed pattern for data transfer.

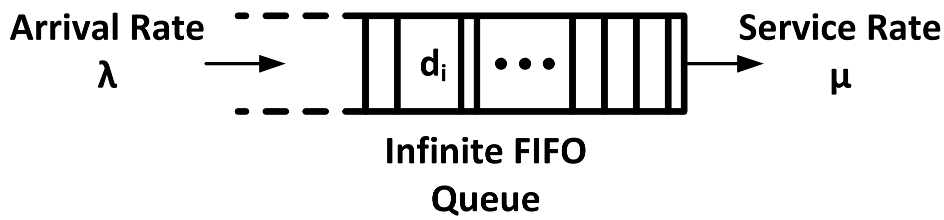


Figure 2.6: Illustration of a queue.

2.4 Energy Efficiency Modeling Using Queues

Use of queues to model communication networks is very common [30]. The simplest queue consists of one server where customers arrive at random time intervals and are serviced with randomly distributed service times. For our purpose, an I/O interface can be considered as a server where data transfer requests arrive at random time intervals. The size of these data requests can also be random which decides the service time for the I/O interface. If the I/O interface is busy transferring data, the data transfer request joins the queue and waits for its turn. For simplicity the queue is assumed to be of infinite capacity and of first in first out (FIFO) nature. This queue is illustrated in Fig. 2.6. Server utilization for such a queue is defined by [30]

$$\text{Utilization}\,\rho = \frac{\text{Mean Arrival Rate}\,(\lambda)}{\text{Mean Service Rate}\,(\mu)} \tag{2.5}$$

2.4.1 Simulation Setup

To model the power consumption of the link, detailed block-wise break-up of power consumption for a 6.25 Gb/s link fabricated in 90 nm CMOS process as reported in [10] is used. Appropriate power scaling is used to model power consumption for DVFS link. Exponential probability distribution functions are used for both inter-arrival time and service time corresponding to an M/M/1 type of queue.

Considering the low voltage circuit limitations, the minimum operating frequency of DVFS

I/O interface was constrained to:

$$\frac{\text{Data Rate}_{\text{min}}}{\text{Data Rate}_{\text{max}}} = \frac{1}{5} \tag{2.6}$$

For the ROO link following parameters were used:

$$\text{P}_{\text{idle}} = 0.01 \times \text{P}_{\text{active}} \tag{2.7}$$

$$\text{Exit Latency} = \frac{10 \times \text{Mean Data Transfer Size}}{\text{Data Rate}_{\text{max}}} \tag{2.8}$$

2.4.2 Results and Discussion

Various energy efficiency metrics were calculated from the definitions given in Section 2.3. In addition, we define energy-delay product (EDP) for an I/O interface as:

$$\text{EDP} = \text{Energy per bit} \times \text{Mean waiting time} \tag{2.9}$$

This metric captures the latency behavior of the I/O interface under randomly distributed data transfer requests. Another interesting point to note is when inter-arrival and service times are constant, the EDP metric resembles BETR (η_{BTM}).

Energy per bit (η_{FTM}), ETR (η_{MTM}), BETR (η_{BTM}), and EDP metrics for always-on, DVFS as well as ROO interfaces are plotted as function of link utilization in Fig. 2.7, Fig. 2.8, Fig. 2.9, and Fig. 2.10, respectively. In the plots, for the curves labeled as DVFS_{EpB} and DVFS_{EDP}, the DVFS data rate is chosen that results in minimum energy per bit and minimum EDP respectively. It is seen that for the energy efficiency metrics that measure performance in terms of average energy consumption and average throughput, DVFS interface optimized for low energy per bit operation provides most efficient operation over a wide range of fractional I/O interface utilization. On the other hand, the EDP metric that accounts for latency of the interface is very poor for DVFS I/O interface optimized for energy per bit performance. Another observation is that due to limited scaling range of DVFS I/O interface (assumed to be 5X in simulation), its energy efficiency degrades a lot at

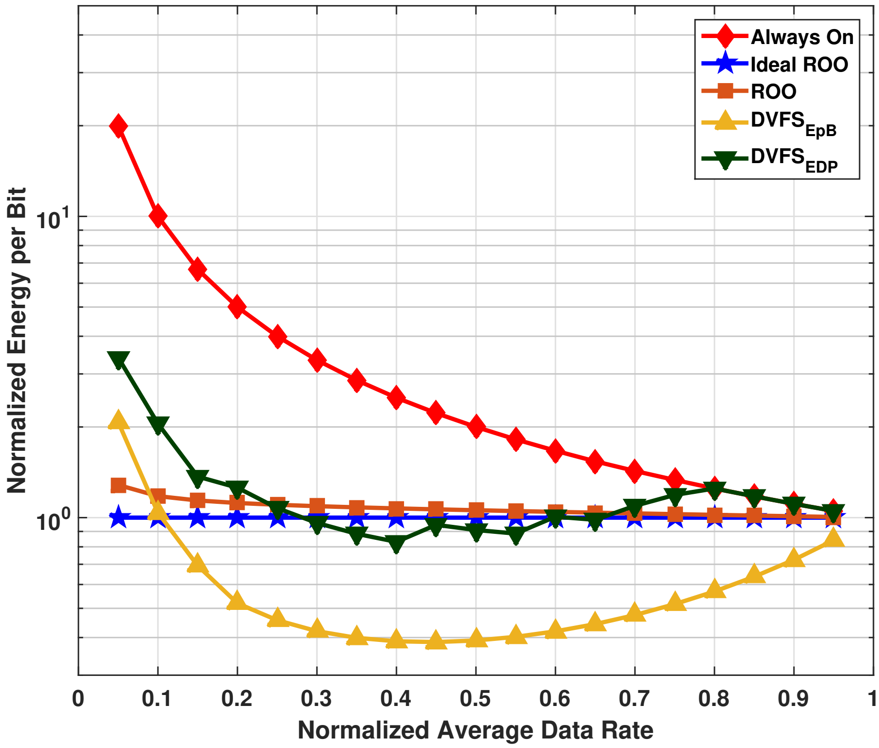


Figure 2.7: Energy per bit for various links from M/M/1 queue simulation.

very low utilization levels. ROO interface, on the other hand, performs equally well for both average energy efficiency metrics as well as EDP metric provided it has low latency response to data transfer requests. Therefore we may conclude that ROO interface is a better choice for latency sensitive applications.

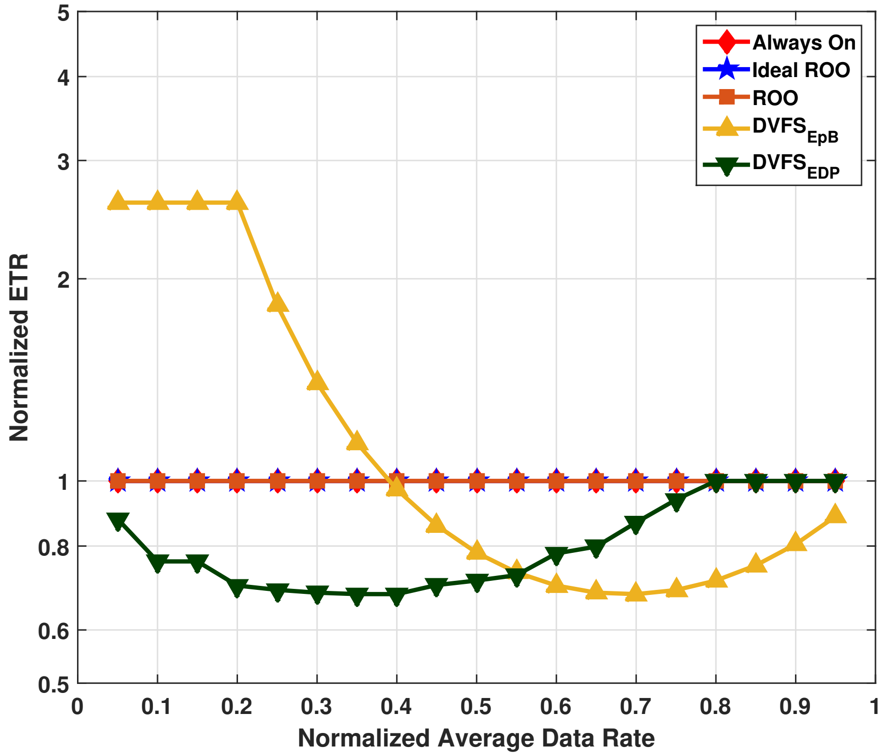


Figure 2.8: ETR for various links from M/M/1 queue simulation.

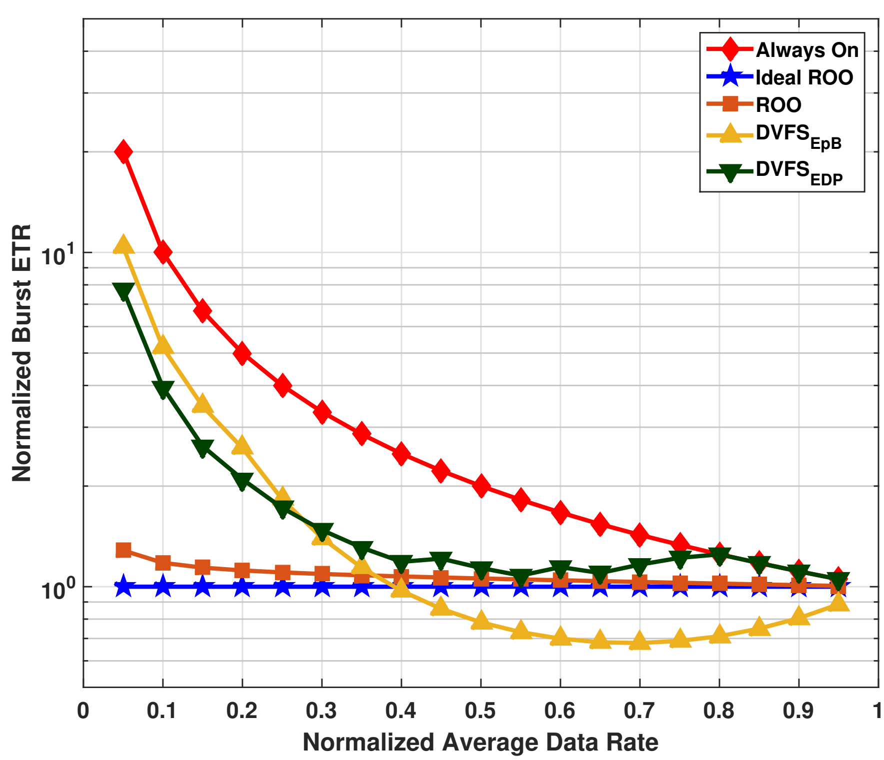


Figure 2.9: Burst ETR for various links from M/M/1 queue simulation.

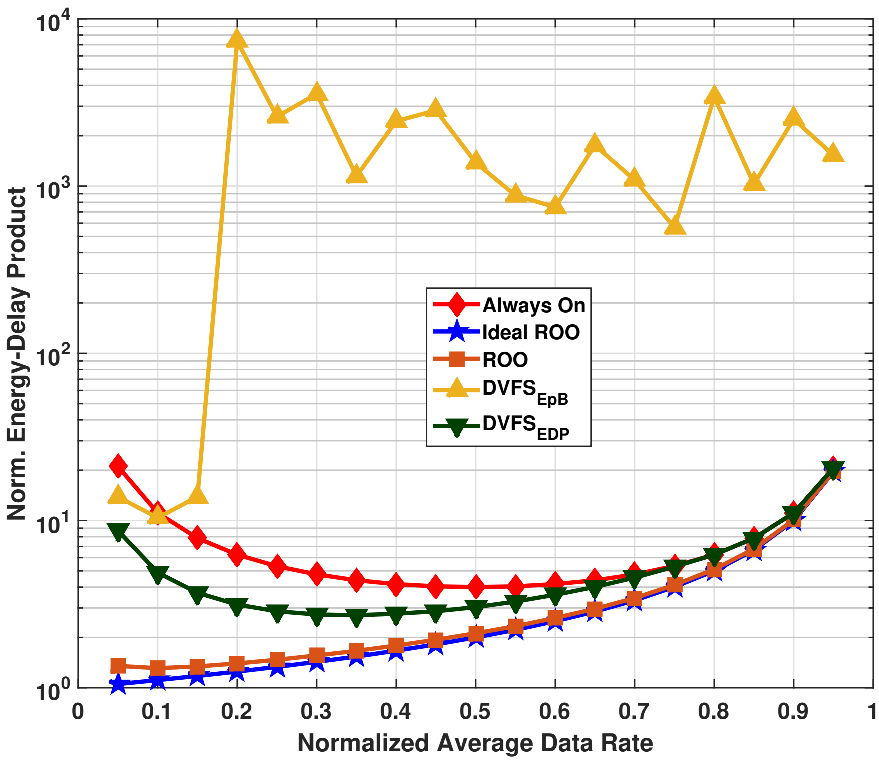


Figure 2.10: Energy-delay product for various links from M/M/1 queue simulation.

CHAPTER 3

DESIGN OF A RAPID ON-OFF TRANSMITTER

As discussed in Chapter 2, it is desirable to scale the power consumption of an I/O interface according to its data rate over a wide range of data transfer bandwidth requirements. DVFS I/O interfaces [19, 20] and ROO I/O interfaces [22, 23] have been proposed to scale power consumption based on data bandwidth requirement for energy efficient operation. While DVFS I/O interfaces save power by scaling supply voltage with data rate, ROO I/O interfaces make use of low power inactive states for saving power when the I/O interface is idle. How quickly an I/O interface can respond to changes in data bandwidth requirements has great influence on the amount of energy that can be saved by using one of the above methods. DVFS I/O interfaces have longer response time due to supply voltage regulators. On the other hand, ROO I/O interfaces are more suitable to track rapid changes in data bandwidth requirements. In many applications the data bandwidth requirements vary across a wide range within very short time intervals. In such cases ROO I/O interfaces can provide more energy savings compared to DVFS I/O interfaces.

In this chapter, we describe techniques for the design of an 8 Gb/s transmitter with a fast turn-on 4 GHz clock multiplier [31] that can be used in a ROO I/O interface. The prototype chip demonstrating these techniques is fabricated in 90 nm CMOS process. When operated with 6 ns turn-on latency and 500 $\mathrm{mV_{ppd}}$ swing, the transmitter energy efficiency varies from 1.7 mW/Gb/s to 2.7 mW/Gb/s for average data rate varying from 8 Gb/s to 64 Mb/s. Under the same conditions, the clock multiplier energy efficiency varies from 0.5 mW/Gb/s/lane to 1.5 mW/Gb/s/lane. The key enablers for this performance are (a) use of a multiplying delay-locked loop (MDLL) as clock multiplier, (b) a fast frequency settling oscillator, and (c) a rapid on-off biasing circuit.

This chapter is organized as follows. Expressions for energy per bit efficiency metric of

ROO I/O interfaces are derived in Section 3.1. Top level transmitter architecture and fast turn-on clock multiplier architecture are described in Section 3.2 and Section 3.3, respectively. Details of clock multiplier implementation and output driver implementation are given in Section 3.4 and Section 3.5, respectively. Section 3.6 discusses the impact of supply and temperature drift on transmitter performance. Measurement results are given in Section 3.7 followed by a description of analytical BER computation technique in Section 3.8.

3.1 Energy Efficiency of ROO I/O Interfaces

We begin with detailed energy efficiency modeling for ROO I/O interfaces. For simplicity we use energy per bit as the energy efficiency metric of interest. Under realistic workload conditions, the peak data rate offered by I/O interfaces is not always fully utilized. Periods of activity are interspersed with idle periods during which the I/O interface does not transfer any data. During the idle period the power consumed by the interface degrades its energy efficiency. Figure 3.1 depicts how the energy efficiency of an always-on interface degrades as its utilization (measured as average data rate) goes below 100%. Behavior of an always-on interface is illustrated in Fig. 3.2 using a hypothetical data transfer pattern. The always-on interface wastes power during the idle time period shown as the shaded region in the figure. An ideal on-off interface consumes power only when it is transferring data and does not consume any power when there is no data transfer activity. But practical interface circuitry cannot be turned on immediately when required. It requires some time to turn on and to turn off. When idle periods are shorter, practical I/O interfaces cannot be turned off without incurring performance penalty due to increased latency. Additionally, as illustrated in Fig. 3.2, a practical interface consumes power not only while turning on and turning off but also while it is not transferring any data. We denote power consumed by an interface when it is not transferring any data as idle power, P_{idle}, whereas active power, P_{active}, is the power consumed by the interface when it is transferring data. The power consumed over and above P_{idle} during the turn-on transient integrated over the turn-on duration is denoted as turn-on transition energy, $E_{tran,on}$. Similarly additional energy consumed during the turn-off transition is denoted as turn-off transition energy, $E_{tran,off}$. The total transition energy spent

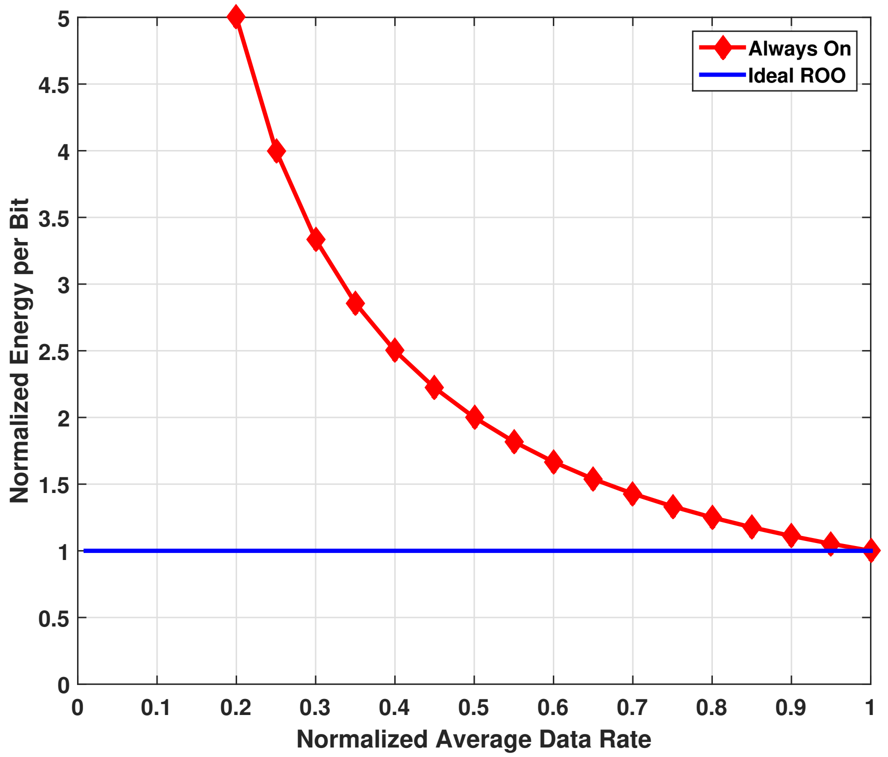


Figure 3.1: Theoretical normalized energy per bit as function of average data rate for I/O links.

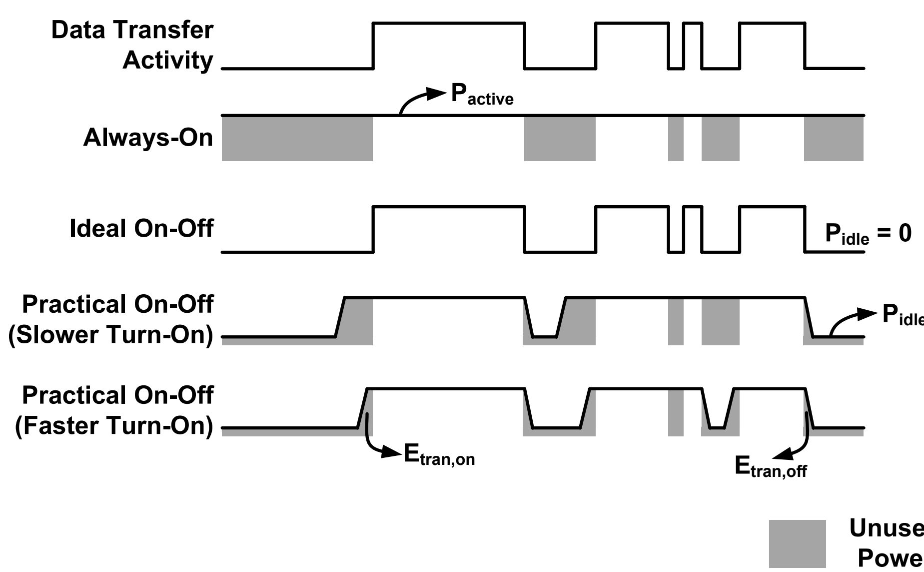


Figure 3.2: Illustration of power consumption pattern for always-on and on-off links for an example data transfer activity.

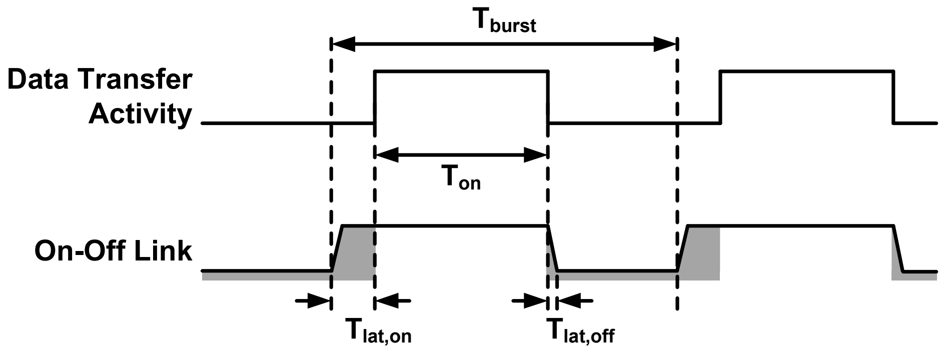


Figure 3.3: A simple repetitive data transfer burst pattern.

in each on-off transition of the interface is

$$\mathrm{E_{tran,tot} = E_{tran,on} + E_{tran,off}} \tag{3.1}$$

Let $\mathrm{T_{lat,on}}$ and $\mathrm{T_{lat,off}}$ be the time taken by the interface to turn on and turn off respectively. Then the total on-off latency (also called as exit latency) of the interface is given by

$$\mathrm{T_{lat,on\text{-}off} = T_{lat,on} + T_{lat,off}} \tag{3.2}$$

The total transition energy consumption is the result of power consumed by various circuit blocks while they approach steady state operating point. To a first order approximation, $\mathrm{E_{tran,tot}}$ is proportional to both active power ($\mathrm{P_{active}}$) of the interface as well as its total on-off latency ($\mathrm{T_{lat,on\text{-}off}}$), i.e.

$$\mathrm{E_{tran,tot} \propto T_{lat,on\text{-}off} P_{active}}$$

For a fixed data transfer burst pattern shown in Fig. 3.3, the average power consumption of an on-off interface is given by

$$\mathrm{P_{avg}} = \left(\frac{\text{Avg. Data Rate}}{\text{Peak Data Rate}}\right)\mathrm{P_{active}} + \frac{\mathrm{E_{tran,tot}}}{\mathrm{T_{burst}}} + \left(1 - \frac{\text{Avg. Data Rate}}{\text{Peak Data Rate}}\right)\mathrm{P_{idle}} \tag{3.3}$$

where

$$\text{Avg. Data Rate} = \frac{\mathrm{T_{on}}}{\mathrm{T_{burst}}} \times \text{Peak Data Rate} \tag{3.4}$$

If we assume

$$\mathrm{E}_{\text{tran,tot}} \approx \mathrm{T}_{\text{lat,on-off}}\mathrm{P}_{\text{active}}$$

then

$$\begin{aligned}
\mathrm{P}_{\text{avg}} &= \left(\frac{\text{Avg. Data Rate}}{\text{Peak Data Rate}}\right)\mathrm{P}_{\text{active}} + \left(\frac{\mathrm{T}_{\text{lat,on-off}}}{\mathrm{T}_{\text{burst}}}\right)\mathrm{P}_{\text{active}} + \left(1 - \frac{\text{Avg. Data Rate}}{\text{Peak Data Rate}}\right)\mathrm{P}_{\text{idle}} \\
&= \left(\frac{\mathrm{T}_{\text{on}} + \mathrm{T}_{\text{lat,on-off}}}{\mathrm{T}_{\text{burst}}}\right)\mathrm{P}_{\text{active}} + \left(\frac{\mathrm{T}_{\text{burst}} - \mathrm{T}_{\text{on}}}{\mathrm{T}_{\text{burst}}}\right)\mathrm{P}_{\text{idle}} \qquad (3.5)
\end{aligned}$$

The energy per bit (EpB) for such an interface is given by

$$\mathrm{EpB}_{\text{tot}} = \left(1 + \frac{\mathrm{T}_{\text{lat,on-off}}}{\mathrm{T}_{\text{on}}}\right)\mathrm{EpB}_{\text{active}} + \left(\frac{\mathrm{T}_{\text{burst}}}{\mathrm{T}_{\text{on}}} - 1\right)\mathrm{EpB}_{\text{idle}} \qquad (3.6)$$

where

$$\begin{aligned}
\mathrm{EpB}_{\text{active}} &= \frac{\mathrm{P}_{\text{active}}}{\text{Peak Data Rate}} \\
\mathrm{EpB}_{\text{idle}} &= \frac{\mathrm{P}_{\text{idle}}}{\text{Peak Data Rate}}
\end{aligned}$$

From Eq. 3.5, it can be seen that the average power consumption of an on-off interface can be scaled with its on time T_{on}. It is also affected by on-off latency of the interface and its idle state power consumption. Figure 3.4 and Fig. 3.5 depict the energy per bit of an on-off interface for varying idle power (P_{idle}) and varying total on-off latency ($\mathrm{T}_{\text{lat,on-off}}$) respectively, in Eq. 3.6. It can be seen that compared to an always-on interface, on-off interface offers linear power scaling and constant energy per bit across a range of average data rates. It can also be observed that the increase in $\mathrm{T}_{\text{lat,on-off}}$ or P_{idle} reduces the energy efficiency of the interface. This degraded energy efficiency is especially prominent for very low average data rate.

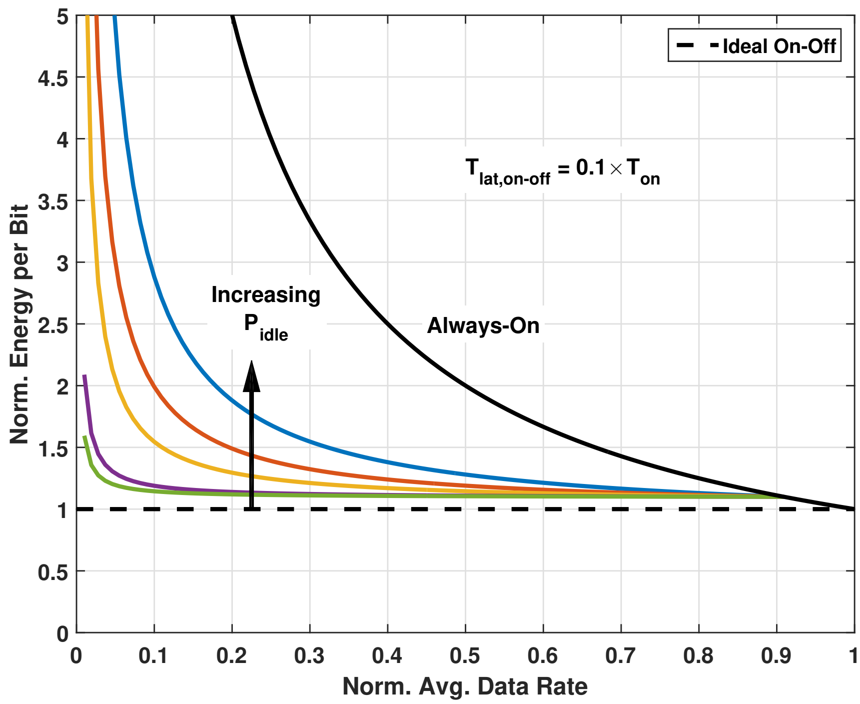


Figure 3.4: On-off link energy efficiency for varying idle power P_{idle}.

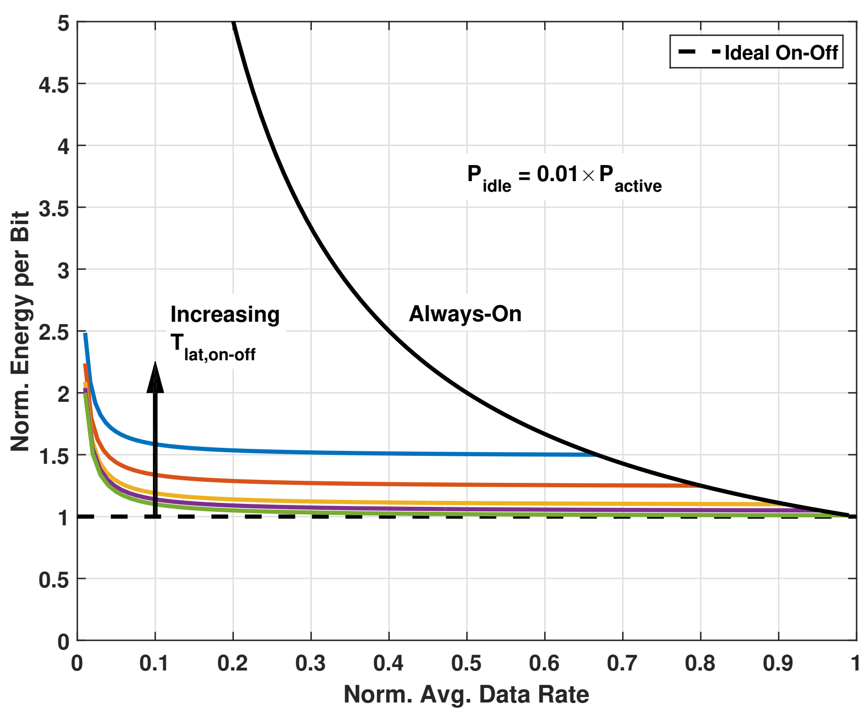


Figure 3.5: On-off link energy efficiency for varying total on-off latency $T_{lat,on-off}$.

3.2 Transmitter Architecture

Clearly, the effectiveness of the on-off interface is limited by finite turn-on latency. Conventional I/O interfaces have longer turn-on time, caused mainly by a slow settling clock generator which is typically implemented using a PLL [22]. Due to its limited bandwidth, a typical PLL has a turn-on time of tens of reference periods at best. Another reason for longer turn-on time is slow settling bias circuitry. To circumvent these issues, work presented in [22] trades off the idle state power saving with link turn-on time using multiple low power states. These multiple low power states are realized by selectively turning off some or all of the constituent circuit blocks during idle time. Rapid on-off transceivers seek to eliminate this trade-off by achieving very short transition time from complete power-off-state. This approach greatly simplifies the interface operation while still allowing energy efficient operation across wide range of data transfer requirements. The present work proposes circuit techniques that can be used to realize a rapid on-off transmitter with a fast turn-on clock multiplier for a rapid on-off interface.

The proposed transmitter architecture is shown in Fig. 3.6. A digital multiplying delay-locked loop (MDLL), used as fast locking clock multiplier, generates 4 GHz clock output MCK from 500 MHz input reference (REF). Power down signals PDNB and $PDNB_{TX}$ are used to turn off the MDLL and output driver, respectively. A 2:1 serializer (SER) combines two 4 Gb/s data inputs (D_{ODD}, D_{EVEN}) to create three, 8 Gb/s inputs ($D_{-1}, D_0,$ and D_1) for a CML output driver with 3-tap feed-forward equalization (FFE). The current source bias voltage (VB) for the CML output driver is obtained using a rapid on-off bias (ROOB) circuit. ROOB circuit overcomes the trade-off between power consumption and area of conventional biasing circuits and its implementation details are discussed in Section 3.5.3.

3.3 MDLL Architecture

Clock multiplier is an important part of a transmitter and has a profound effect on the transmitter performance. It is required to generate a clean high frequency clock with which the transmitter retimes the output data. A phase-locked loop (PLL) based clock multiplier

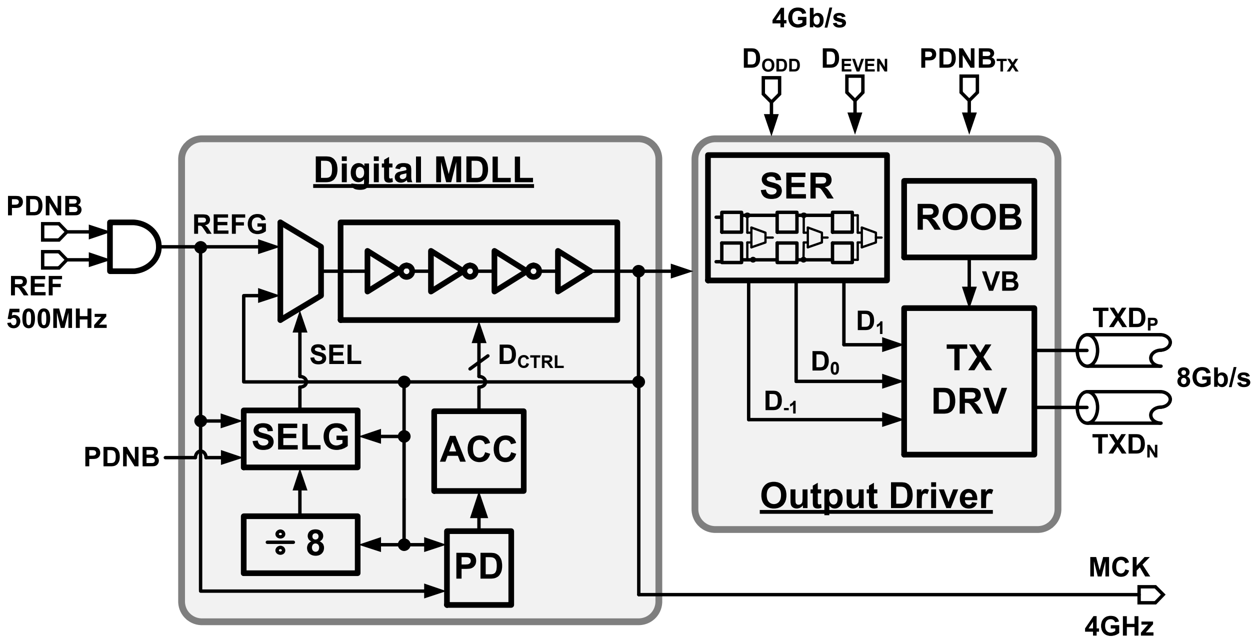


Figure 3.6: Block diagram of the proposed rapid on-off transmitter.

is commonly used in high speed link designs. In case of a ROO transmitter, the clock multiplier is required to stop the clock generation and restart it rapidly without any loss in jitter performance. Considerations of stability put a limit on PLL bandwidth and therefore result in slow phase acquisition of PLL when it restarts. Conventional PLLs have settling time of the order of around $100\,T_{REF}$, where T_{REF} is the reference clock period [32, 22]. Fast lock acquisition techniques for PLL such as gear-shifting [33] and bandwidth adaptation [34] have been explored before and can reduce the PLL acquisition time to the order of 10's of T_{REF} at the expense of added complexity. More importantly these studies do not explore the power cycling behavior of the PLL circuitry and its effect on lock acquisition. Additionally these methods do not assume any *a priori* information about the input reference frequency before the beginning of lock acquisition. Multiplying injection locked oscillators (MILOs) are shown to have much shorter acquisition time ($< 10\,T_{REF}$) and have been used in fast-hopping frequency synthesizers [35]. In [23], a MILO-based clock multiplier is demonstrated to turn-on within 6 reference cycles while being power cycled. A fixed reference frequency is assumed in [23], whereas the work in [36] extends the use of fast-locking MILO technique for wide range of input reference frequencies.

While PLLs suffer from slow phase acquisition, it is relatively easy to achieve low reference spurs and hence low pattern jitter on clocks generated using PLLs. On the other hand, there exists a trade-off between acquisition time and output reference spurs for MILO [35]. Therefore faster settling time for MILO typically results in larger pattern jitter on the output clock. To find an optimal alternative to either PLL or MILO, it is useful to revisit the reason for slower phase acquisition of conventional PLLs. The slower phase acquisition of conventional PLLs is a result of two important factors : (a) initial frequency relationship between the reference input and the PLL oscillator is considered unknown, (b) initial phase relationship between reference input and PLL oscillator is considered unknown. Any error in initial condition when PLL starts up, results in settling transients in the loop which are governed by the loop bandwidth. Interestingly, the work in [37] demonstrates much faster turn-on time, albeit for a fixed duty cycle ratio, for a PLL where oscillator frequency is matched to reference frequency after initial locking and stays the same for burst mode operation. A fixed phase relationship between reference clock and PLL output is maintained indirectly unlike the direct comparison that occurs in conventional PLLs.

In this work, we utilize multiplying delay-locked loop (MDLL) as a fast-locking clock multiplier for ROO transmitter. MDLL has previously been proposed as an alternative to PLL for clock multiplication [38]. Figure 3.7 shows the block diagram of a typical MDLL. It consists of a multiplexed ring oscillator in which the ring oscillator loop can be broken using a multiplexer to pass a reference signal instead. The select signal for the multiplexer is generated by select pulse generation (SELG) block. A loop formed by phase detector (PD), charge pump (CP), and loop filter (LF) generates control voltage, V_{CTRL}, for the multiplexed ring oscillator. MDLLs enjoy many advantages over PLLs such as reduced phase noise accumulation and low power supply sensitivity [39]. These advantages are derived by selectively replacing oscillator edge with the reference edge which resets the oscillator phase every reference clock cycle. Oscillator phase reset caused by this selective edge replacement can be used to establish a fixed phase relationship between oscillator phase and reference input, thereby alleviating slow phase settling transients common in conventional PLLs. Frequency locking on the other hand can be achieved by using a slower initial settling transient to establish a correct value of V_{CTRL}. If this V_{CTRL} can be maintained for subsequent power cycling operations, instantaneous locking can be achieved when MDLL turns back on.

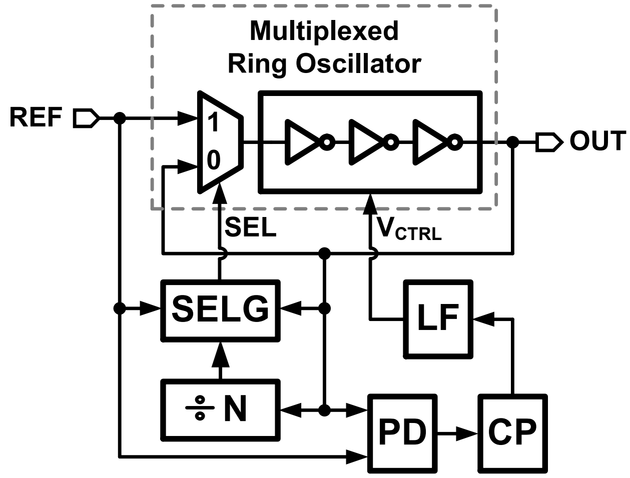


Figure 3.7: Conventional MDLL block diagram.

To illustrate the fast locking using MDLL, consider the timing diagrams shown in Fig. 3.8. Initial slow locking transient establishes the oscillator control voltage V_{CTRL} such that $NT_{OSC} = T_{REF}$, where N is the frequency multiplication factor and T_{OSC} is oscillator time period. The SELG block generates SEL signal once every reference cycle to pass the positive edge of the reference to the ring oscillator. In the absence of reference edge, the SELG block waits for positive edge on the reference (REF). As the ring oscillator loop is broken in this condition, the oscillator output OUT stops oscillating and remains at a fixed voltage level. The MDLL can be considered off under these circumstances. As soon as positive edge on REF arrives, it is propagated through the ring oscillator delay chain and the SELG block acts to close the ring oscillator loop. If the ring oscillator frequency is restored to exactly the same level where it was during steady state operation, the MDLL achieves instantaneous phase lock after the REF edge. Restoring both phase and frequency of the ring oscillator to known conditions when the MDLL turns back on contributes to its instantaneous phase locking.

Using MDLL as a fast-locking clock multiplier decouples the jitter performance of the clock generator from its turn-on time. It can potentially maintain very good jitter performance as well as short turn-on time without any additional steady state power penalty. As will be seen later, the turn-on time of a MDLL can be made equal to the frequency settling time of the multiplexed ring oscillator.

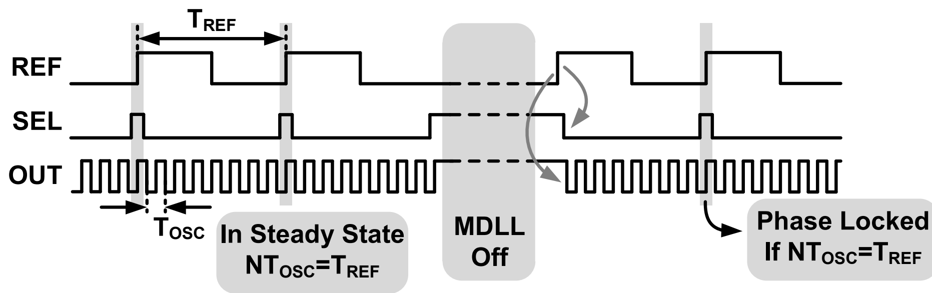


Figure 3.8: Illustration of MDLL on-off behavior using timing diagram.

From the previous discussion it is clear that maintaining frequency information during the off time is important for fast phase-locking of a MDLL. In a conventional analog MDLL [38] (Fig. 3.9) this information is stored on loop filter capacitor, C_{LF}, in the form of control voltage V_{CTRL}. Any leakage on C_{LF} leads to loss of frequency locking information due to change in V_{CTRL} when MDLL is off. This results in increased turn-on time for the analog MDLL as its frequency acquisition is governed by a slow feedback loop. An alternate highly digital implementation of MDLL [40, 39] is shown in Fig. 3.10. In this architecture the feedback loop is formed by time-to-digital converter (TDC), digital accumulator (ACC), digital delta-sigma converter, digital-to-analog converter (DAC), and low pass filter (LPF). The frequency locking information is available in digital format at the output of ACC, which is then converted into analog voltage using DAC and LPF. Settling time required for the LPF limits the turn-on time of this digital MDLL architecture. The use of digital delta-sigma converter and DAC allows much finer frequency resolution of the oscillator while the LPF is necessary to filter the high frequency quantization error shaped by the delta-sigma modulator. Due to their limited jitter accumulation, MDLLs can operate with coarser frequency resolution as compared to PLLs. In such case, use of Nyquist-rate DAC with limited resolution is preferred for fast turn-on MDLL over a combination of delta-sigma modulator, DAC and LPF.

The proposed fast turn-on MDLL architecture is shown in Fig. 3.11. In this digital MDLL architecture, a digitally controlled multiplexed ring oscillator (DXRO) replaces the combination of delta-sigma modulator, DAC, LPF, and multiplexed ring oscillator from the highly digital MDLL architecture shown in Fig. 3.10. Use of DXRO not only results in faster

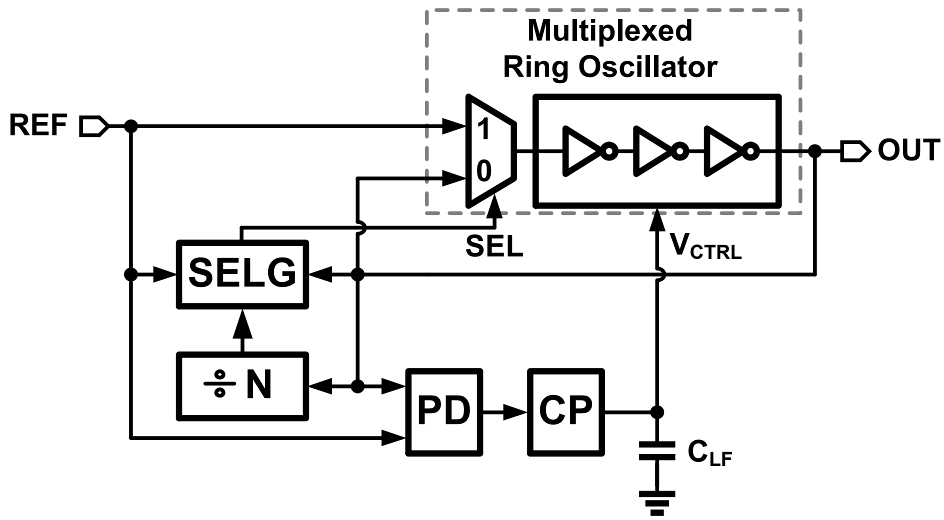


Figure 3.9: Block diagram of a conventional analog MDLL.

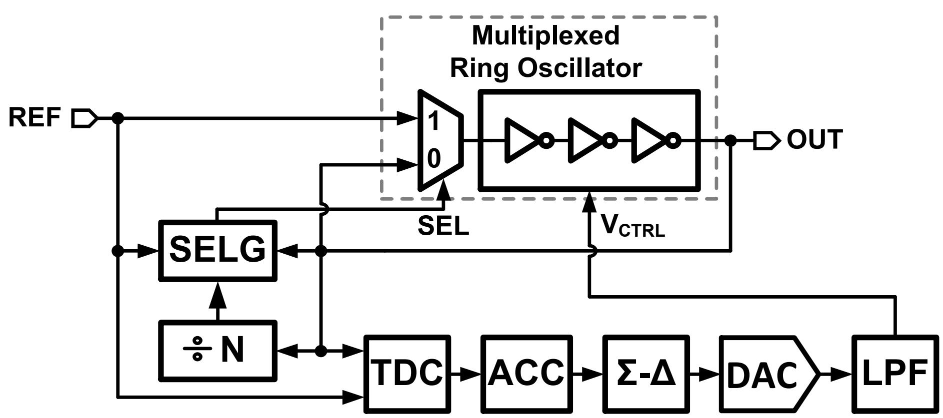


Figure 3.10: Block diagram of a conventional digital MDLL.

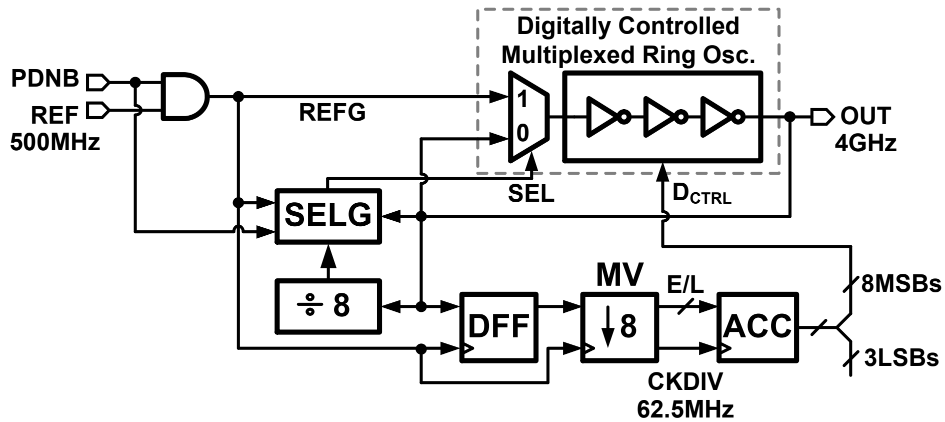


Figure 3.11: Block diagram of the proposed MDLL architecture.

turn-on but also obviates any need for external biasing circuitry. In the proposed MDLL a 500 MHz reference clock (REF) is multiplied by a factor of 8 to generate a 4 GHz output clock (OUT). A power-down signal (PDNB) is used to gate the REF input when MDLL needs to be turned off. It should be noted that the PDNB signal needs to be retimed with respect to the REF input to avoid any glitches on the gated reference clock, REFG. SELG block generates select pulses for DXRO multiplexer using appropriate logic. A D flip-flop (DFF) is used as a bang-bang phase detector for the DXRO tuning loop [39]. The output of DFF is decimated by a factor of 8 using a majority voting block (MV). The early/late outputs from the MV block are fed into an 11-bit digital accumulator (ACC). The ACC block is clocked at much lower frequency of 62.5 MHz which is generated by dividing the reference frequency by 8. This reduces the power consumed in the ACC block at the expense of slower frequency acquisition of the DXRO tuning loop. The 11-bit output of ACC is further truncated by 3 LSBs to reduce the steady state dithering of the ACC output and the resulting output jitter. An 8-bit digital input to the DXRO tunes its frequency to $8F_{\mathrm{REF}}$. To ensure locking of the MDLL, the DXRO is always started from its highest oscillation frequency.

To illustrate the rapid on-off operation using proposed architecture, simulated waveforms for important signals are shown in Fig. 3.12. As shown, the digital control word for frequency tuning is set during the initial frequency acquisition. After the slow initial settling, the subsequent on-off operations can happen very rapidly with gating of the input reference clock. Figure 3.13 shows the simulated waveforms during the MDLL turn-on transient. When the

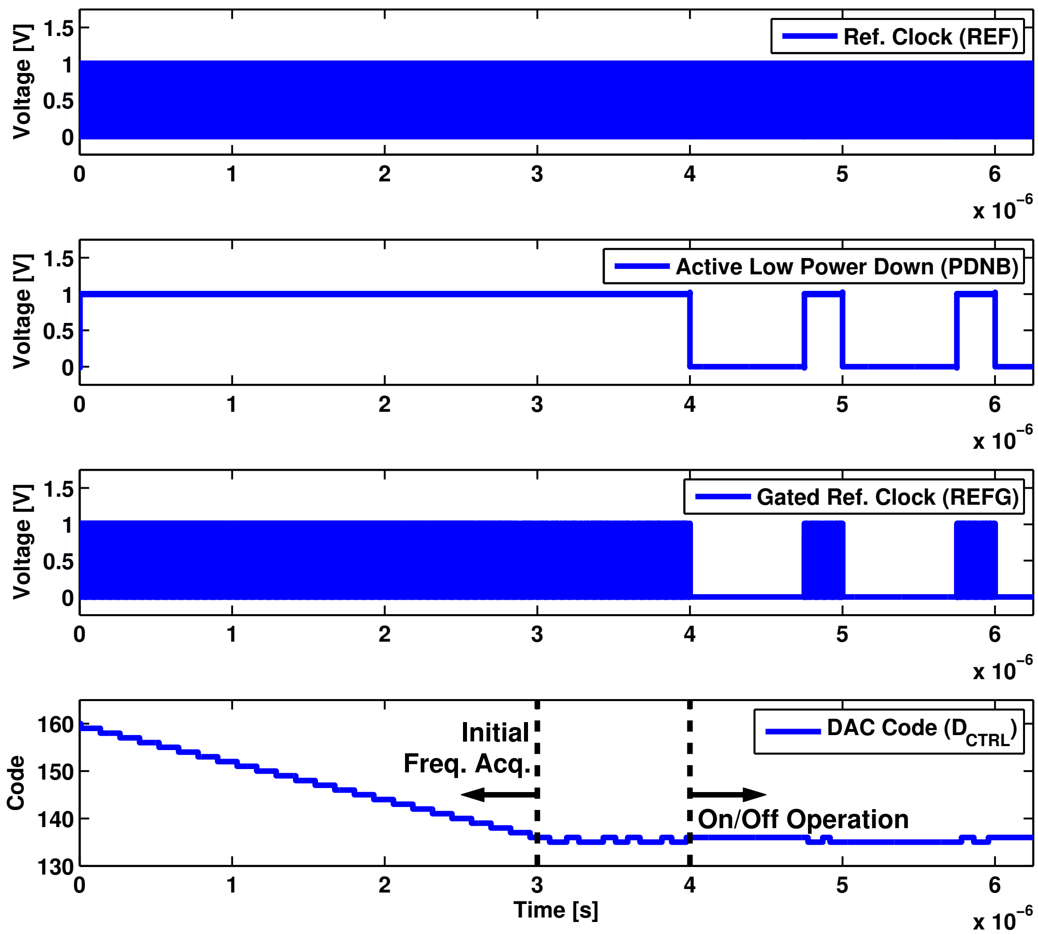


Figure 3.12: Simulated transient waveforms for proposed MDLL architecture.

PDNB signal is pulled high to remove the reference clock gating, the high frequency clock appears at the MDLL output. From the figure, it is clear that there is a settling transient associated with the output waveform. This transient is mainly decided by frequency settling performance of the DXRO. The simulated MDLL turn-off transient waveforms are shown in Fig. 3.14. It should be noted that the MDLL output waveform turn-off is synchronized to the gated reference input by design and therefore avoids any spurious glitches while turning off.

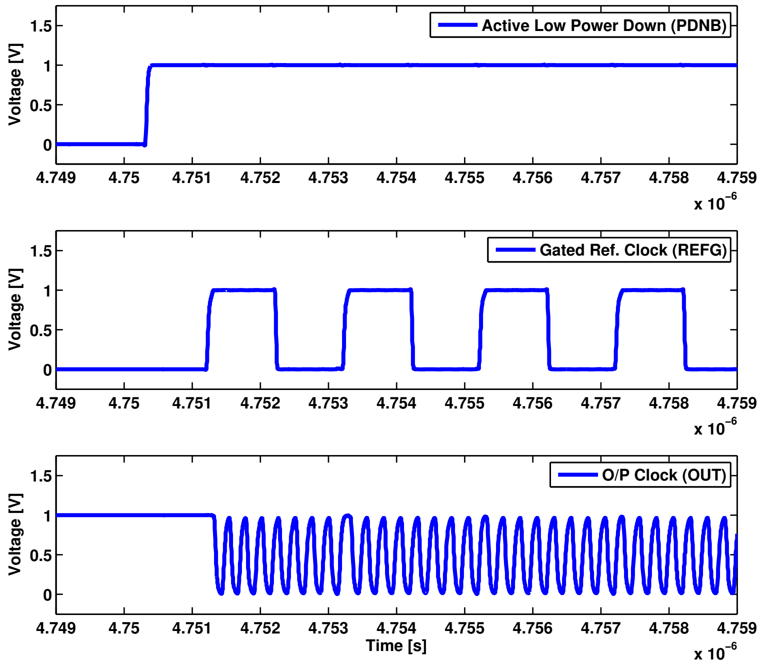


Figure 3.13: Simulated MDLL turn-on transient waveforms.

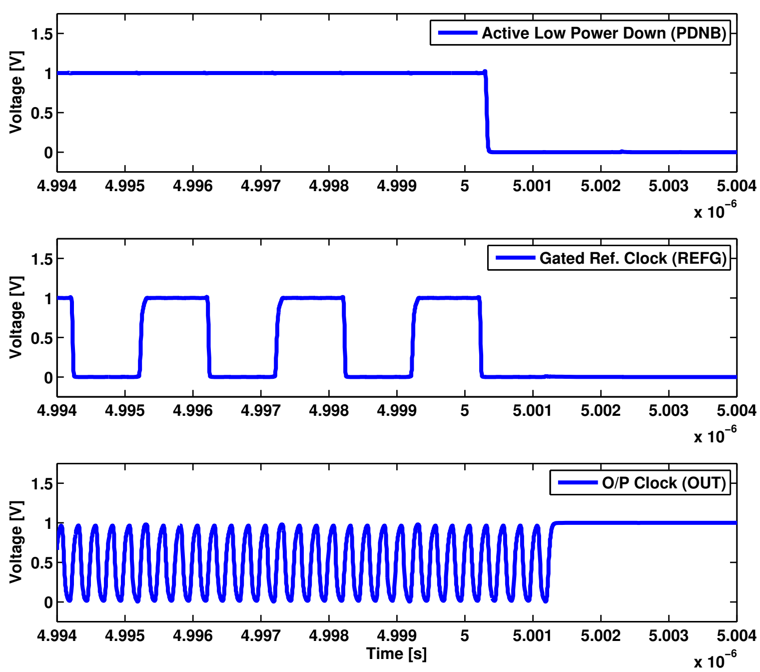


Figure 3.14: Simulated MDLL turn-off transient waveforms.

3.4 MDLL Implementation

3.4.1 Digitally Controlled Multiplexed Ring Oscillator (DXRO)

A digitally controlled oscillator (DCO) used in the fast-locking MDLL design is required to meet three key requirements: (a) fast frequency settling, (b) output state retention in off-state, and (c) no static power consumption in off-state. Although CML-based delay cells provide better power supply rejection and have been used before for fast turn-on oscillators [23], they suffer from higher power consumption as well as increased phase noise due to low swing operation. Additionally, CML-based delay stages require separate fast turn-on biasing and cannot retain the logic state in the off-state. Pseudo-differential CMOS inverter-based delay stages, however, provide larger output swing and better steady state phase noise performance; therefore, they are commonly used in ring oscillators. CMOS inverter-based delay cells can also maintain the logic state while consuming only device leakage power in the absence of input transitions. In view of these advantages, pseudo-differential CMOS inverter-based delay cells are selected for DXRO implementation.

A common method for digital control of frequency is by current starving the ring oscillator delay cells as shown in Fig. 3.15. In Fig. 3.15(a), a voltage-mode DAC is used to generate analog voltage V_{DAC}, which in turn controls current through transistor M_P. When MDLL is off and the ring oscillator delay cells consume no current, M_P goes in the deep triode region while V_{CTRL} node approaches the supply voltage VDD. When MDLL turns back on, the delay cells start drawing current and V_{CTRL} starts going low. Due to the large drain to gate parasitic coupling capacitor, C_C, of M_P in the triode region, any disturbance on V_{CTRL} disturbs V_{DAC} as well. A similar behavior is observed in case of current starved DCO, controlled using current-mode DAC, as shown in Fig. 3.15(b). In this case, bias voltage V_B is disturbed when MDLL turns on. Figure 3.15(c) shows representative voltage waveforms for current starved oscillators illustrating the increased settling time due to the disturbance on V_{DAC} / V_B nodes. To minimize this disturbance, the output impedance of voltage-mode DAC in Fig. 3.15(a) or the bias generator (BIASGEN) in Fig. 3.15(b) needs to be very low. Ensuring low output impedance for these blocks results either in increased

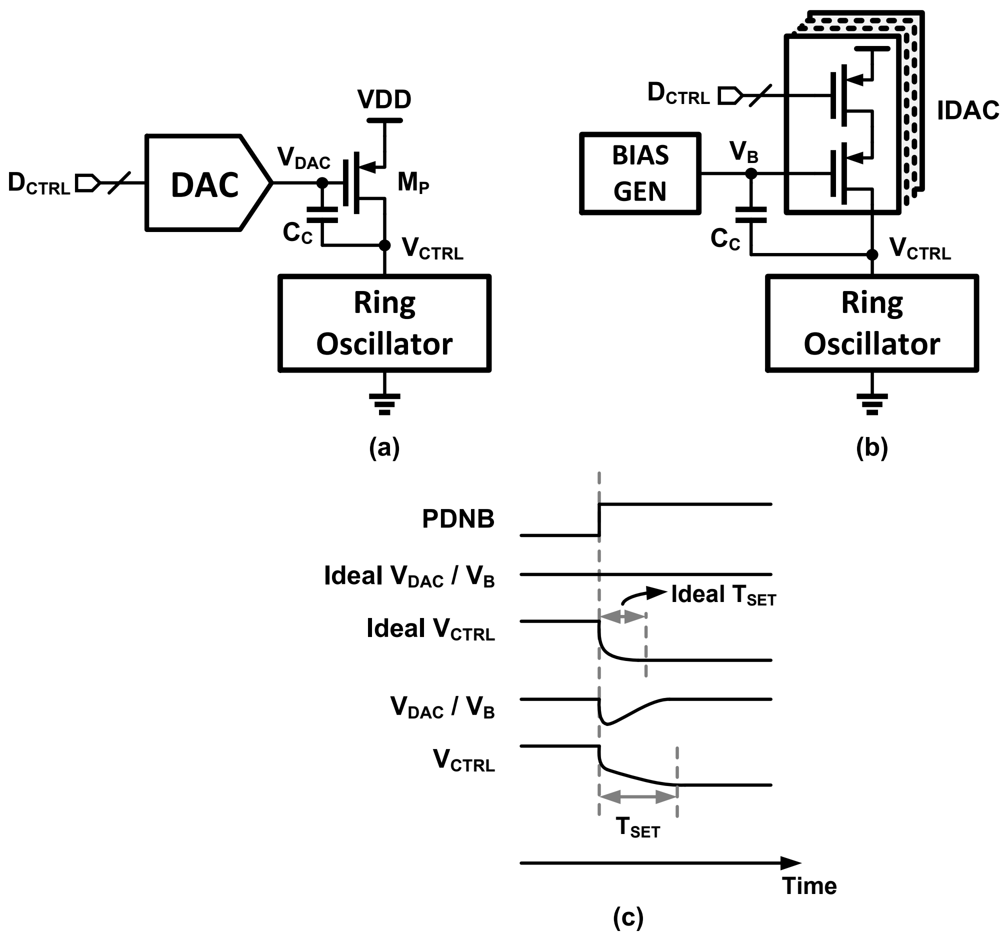


Figure 3.15: (a) Voltage controlled, and (b) current controlled current starved oscillators with (c) illustrative voltage settling waveforms.

power consumption or in use of large decoupling capacitance which may have to be external. In addition to the above mentioned drawback, the current starved oscillator also consumes current in the off-state in either the always-on DAC or the always-on BIASGEN circuit.

The DCO proposed in [41] avoids use of an explicit DAC by varying the strength of delay cells to achieve digital control of frequency. This DCO architecture can exhibit very fast frequency settling behavior as it does not rely on an external analog bias or control voltage. But, to achieve high frequency resolution, this architecture uses a large number of delay cells that act as additional capacitive load increasing the power consumption at high oscillation frequencies. Another DCO architecture proposed in [42] makes use of digitally controlled resistor in the delay cell supply path to control the frequency of the DCO. The digitally controlled resistor does not require any analog bias voltage and hence can result in very fast DCO frequency settling. Additionally, its simple structure makes it amenable for power efficient high frequency operation in the MDLL loop. Therefore this architecture is chosen to implement the DXRO for the fast turn-on MDLL.

Figure 3.16 shows the top level schematic diagram of the DXRO used in the fast turn-on MDLL. DXRO consists of four pseudo-differential CMOS inverter-based delay stages. An even number of stages are chosen so as to generate quadrature output phases, although only in-phase clocks are used in the transmitter. Two out of four delay stages, denoted as variable delay stages in Fig. 3.16, are connected to a variable supply generated using PMOS and NMOS transistor based resistor-mode DACs (RDACs). Delay of the variable delay stages can be tuned using variable supply voltages V_{CP}, and V_{CN}, which in turn are controlled by RDAC control word D_{CTRL}. Use of both PMOS transistor based RDAC (P-RDAC) and NMOS transistor based RDAC (N-RDAC) ensures that the common mode output of the variable delay stage is approximately $V_{DD}/2$, obviating the need for explicit level shifting buffers. Avoiding level shifting buffers not only saves power but also removes any turn-on time associated with the conventional capacitively coupled level shifting buffers. A swing restore stage operating with V_{DD} supply restores the output swing to rail-to-rail levels.

The circuit diagram of the variable delay stage is shown in Fig. 3.17(a). It uses feed-forward resistors (R_F) to achieve pseudo-differential operation. Digitally controlled MOS

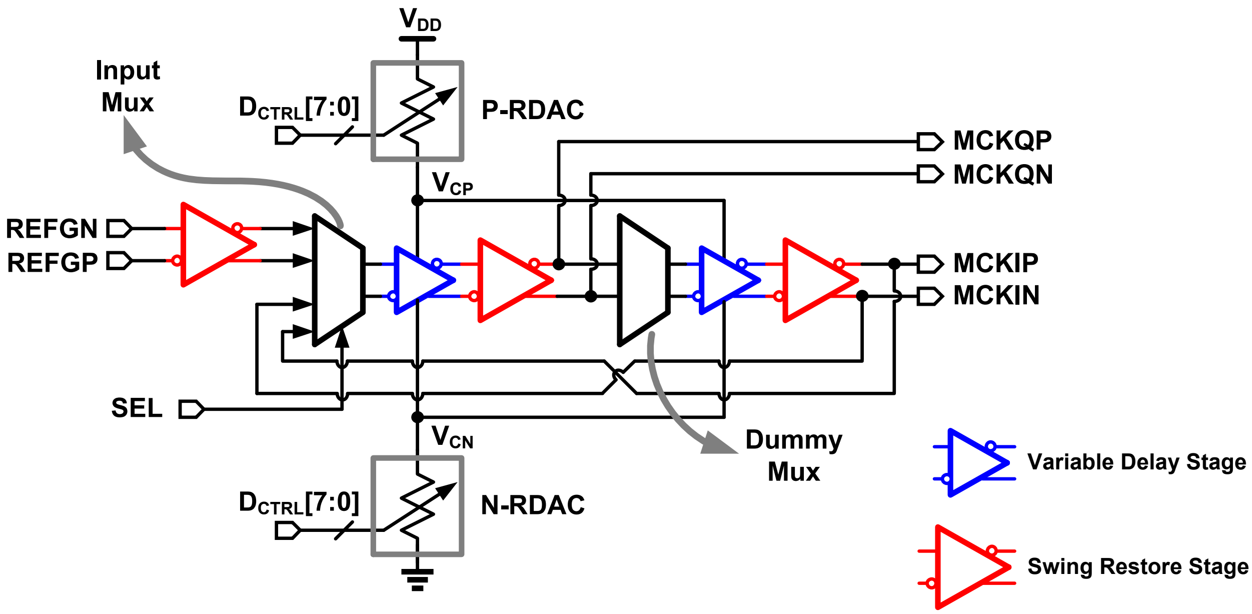


Figure 3.16: Top level schematic diagram of digitally controlled multiplexed ring oscillator (DXRO).

capacitors are used for coarse frequency tuning across process corners with manually tuned 2-bit digital code D_{COARSE}. The fine frequency tuning is done by varying the supply voltages V_{CP}, and V_{CN} using P-RDAC and N-RDAC. Figure 3.17(b) shows the circuit diagram of the swing restore stage. The weak strength back-to-back connected inverters provide pseudo-differential operation as well as rudimentary duty cycle correction of the input waveform generated by the variable delay stages. DXRO delay stages as well as multiplexers utilize low-V_t devices for high frequency operation.

To minimize the reference spurs at the MDLL output, it is important to match the shape of both reference as well as oscillator waveforms that go to the input multiplexer. It is also necessary to reduce the disturbance to the oscillator due to reference edge selection. Figure 3.18 depicts the block diagram at the multiplexer input of the DXRO. An identical swing restore stage is used to pass the reference edge to the input multiplexer. The use of swing restore stage not only matches the reference waveform to oscillator waveform but also avoids any disturbance to variable delay stage supply voltages as the swing restore stage is connected to V_{DD} supply. Input multiplexer is implemented using transmission gates due to their amenability to high frequency operation without consuming voltage headroom of

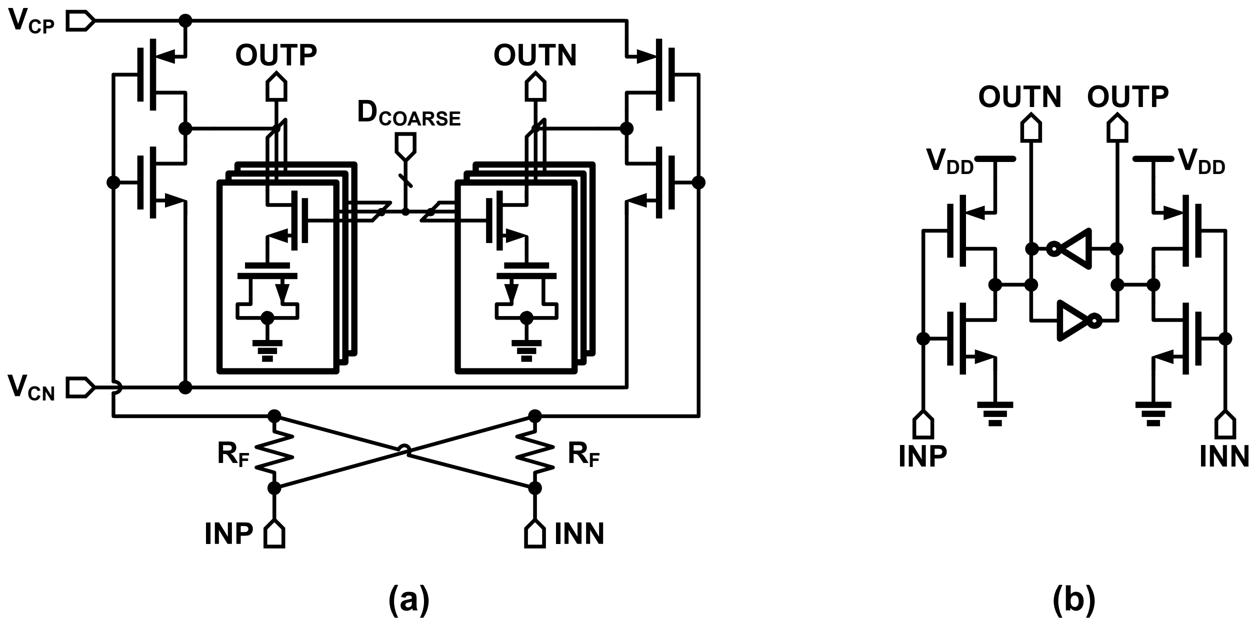


Figure 3.17: (a) Variable delay stage circuit diagram, and (b) swing restore stage circuit diagram.

the delay stages. Use of transmission gate multiplexer also results in the oscillator output (MCKIP, MCKIN) being loaded differently when reference edge is being passed through the multiplexer. To ensure equal loading on the oscillator output all the time, a multiplexer driven with complementary select input is used to connect the oscillator output to a dummy delay cell when reference edge is selected in the multiplexer.

The RDACs used for frequency tuning of the DXRO influence many important MDLL performance metrics such as frequency tuning loop dynamics, frequency tuning range, output jitter due to frequency or, equivalently, period quantization error in addition to the frequency settling time of the DXRO. It can be shown that the deterministic jitter due to period quantization error, $\Delta \mathrm{T_Q}$, is given by

$$\mathrm{T_{DJQ,p\text{-}p}} = N\,\Delta \mathrm{T_Q} \tag{3.7}$$

For an RDAC-based tuning, the DXRO period is approximately given by

$$\mathrm{T_{DXRO}} \approx \mathrm{T_{DXRO,min}} + k\mathrm{R_{RDAC}} \tag{3.8}$$

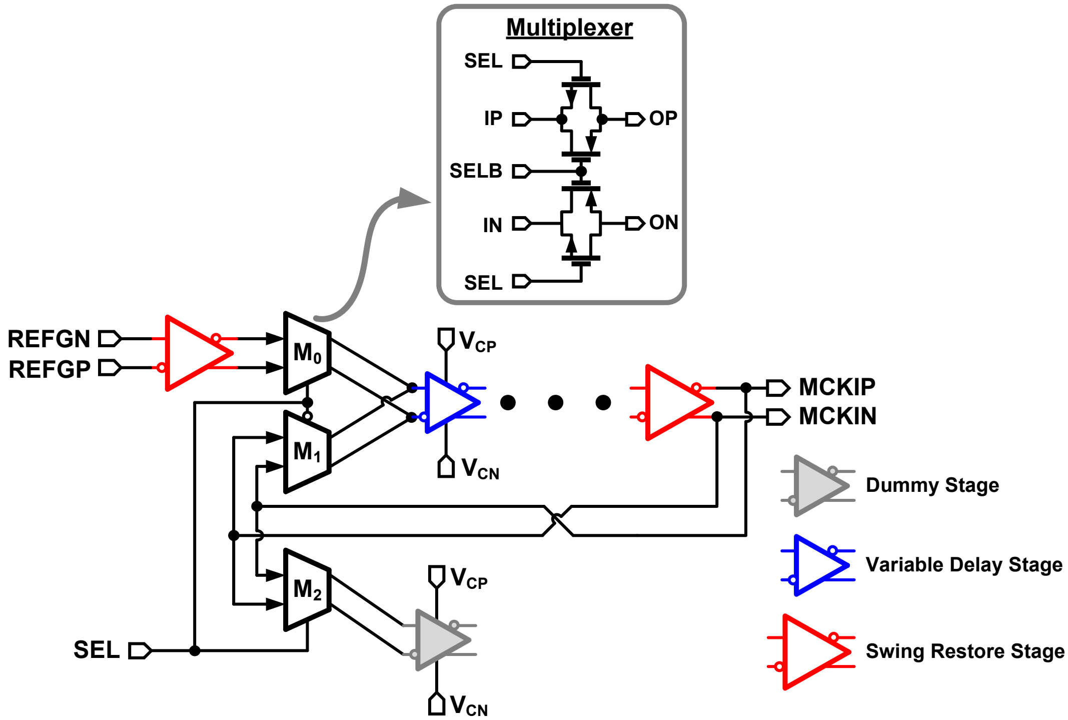


Figure 3.18: DXRO input multiplexer arrangement for matching shape of reference and oscillator voltage waveforms.

where k is a constant of proportionality decided by delay cell circuit parameters and supply voltage, and $\mathrm{R_{RDAC}}$ is the RDAC resistance. Therefore, the period quantization error is proportional to RDAC resistance step size. The minimum and maximum values of RDAC resistance also decide the fine frequency tuning range of the DXRO. Recall that coarse frequency tuning is achieved by a 2-bit capacitor array at the output of the variable delay cell of DXRO. A linear RDAC resistance characteristic with input code is preferred to maintain DXRO deterministic jitter due to quantization error. The MOS transistor resistance in deep triode region is given by

$$\mathrm{R_{on}} = \frac{1}{\mu \mathrm{C_{ox}} \left(\frac{W}{L}\right) (\mathrm{V_{DD}} - \mathrm{V_t})} \tag{3.9}$$

As $\mathrm{R_{on}}$ is inversely proportional to transistor width, a parallel combination of identical MOS transistors results in 1/x resistance characteristic. In [43, 42] a series-parallel combination of MOS transistors is used to achieve linear RDAC resistance characteristic. Use of series connection of MOS resistors entails wider devices and increases area. Figure 3.19 shows the circuit diagrams for PMOS and NMOS transistor based RDACs used in this implementation.

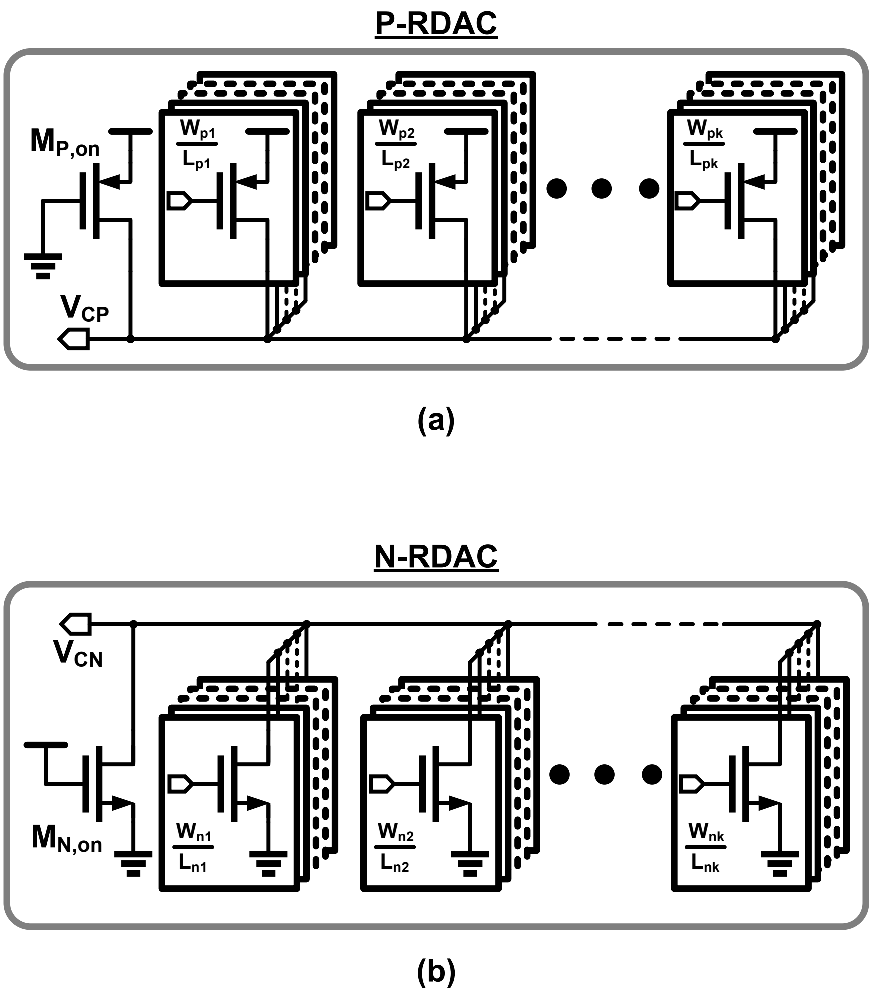


Figure 3.19: (a) PMOS transistor based RDAC, (b) NMOS transistor based RDAC.

As depicted, a parallel MOS combination is used in RDACs to save area. To obtain linear resistance characteristic, the parallel devices are divided in banks with different unit elements. The devices in each bank are identically sized to closely match a linear RDAC transfer characteristic in a narrow code-range. A linear RDAC transfer characteristic over all the codes is obtained by combining piece-wise linear characteristics of individual banks. Thermometer coding is utilized to ensure monotonicity even in the presence of mismatch between unit resistors. The RDAC resistance is equally divided between P-RDAC and N-RDAC. Duty cycle error that can be caused by the mismatch between P-RDAC and N-RDAC resistance values can be corrected by the swing restore stage that follows the variable delay stage.

Both P-RDAC and N-RDAC are designed to provide resistance values going from 1.2 kΩ to

$20\,\Omega$ under typical conditions. To maintain linear period characteristic of the DXRO across this wide range of resistance values, a smaller resistance step size is required at low resistance values. Figure 3.20 shows the simulated resistance characteristic for P-RDAC. As can be seen in Fig. 3.20, the slope of the resistance characteristic is different at very low resistance values compared to high resistance values. The DXRO period as a function of input digital code of the RDAC is shown in Fig. 3.21. Under typical operating conditions the simulated period resolution of the DXRO is seen to be less than 300 fs. This quantization error would result in worst case peak to peak jitter of 2.4 ps. From Fig. 3.21, the simulated frequency tuning range of the DXRO is observed to be from 3.78 GHz to 4.34 GHz, for a single coarse control code. Simulations also indicate that the overall frequency tuning range, obtained by utilizing coarse tuning code ($\mathrm{D_{COARSE}}$), is from 3 GHz to 4.72 GHz under typical conditions.

The RDAC resistance also plays an important role in deciding the frequency settling time of the DXRO. Figure 3.22 illustrates the poles associated with DXRO frequency settling. The simulated transient settling waveforms for the DXRO turn-on event are shown in Fig. 3.23. When DXRO is turned off, the voltages on nodes $\mathrm{V_{CP}}$ and $\mathrm{V_{CN}}$ reach the supply and the ground voltages respectively. When DXRO is turned on by the reference edge, the DXRO starts oscillating at higher frequency due to higher voltage difference between $\mathrm{V_{CP}}$ and $\mathrm{V_{CN}}$. With DXRO oscillations, $\mathrm{V_{CP}}$ and $\mathrm{V_{CN}}$ nodes start settling with time constants given by

$$\tau_{\mathrm{P}} = \mathrm{R_{RDAC,P}C_{PAR,P}} \tag{3.10}$$

$$\tau_{\mathrm{N}} = \mathrm{R_{RDAC,N}C_{PAR,N}} \tag{3.11}$$

where $\mathrm{R_{RDAC,P}}$ and $\mathrm{R_{RDAC,N}}$ are resistances of P-RDAC and N-RDAC respectively, while $\mathrm{C_{PAR,P}}$ and $\mathrm{C_{PAR,N}}$ are parasitic capacitances on nodes $\mathrm{V_{CP}}$ and $\mathrm{V_{CN}}$ respectively. The initial faster DXRO oscillations result in input multiplexer select signal, SEL, going high earlier than its ideal location. Consequently the DXRO waits longer for reference input edge. Due to lack of transitions while waiting for the reference edge, DXRO current consumption reduces. As seen from the zoomed-in part of Fig. 3.23, the $\mathrm{V_{CP}}$ node voltage climbs up and $\mathrm{V_{CN}}$ node voltage goes down while the DXRO is waiting for reference edges. Longer

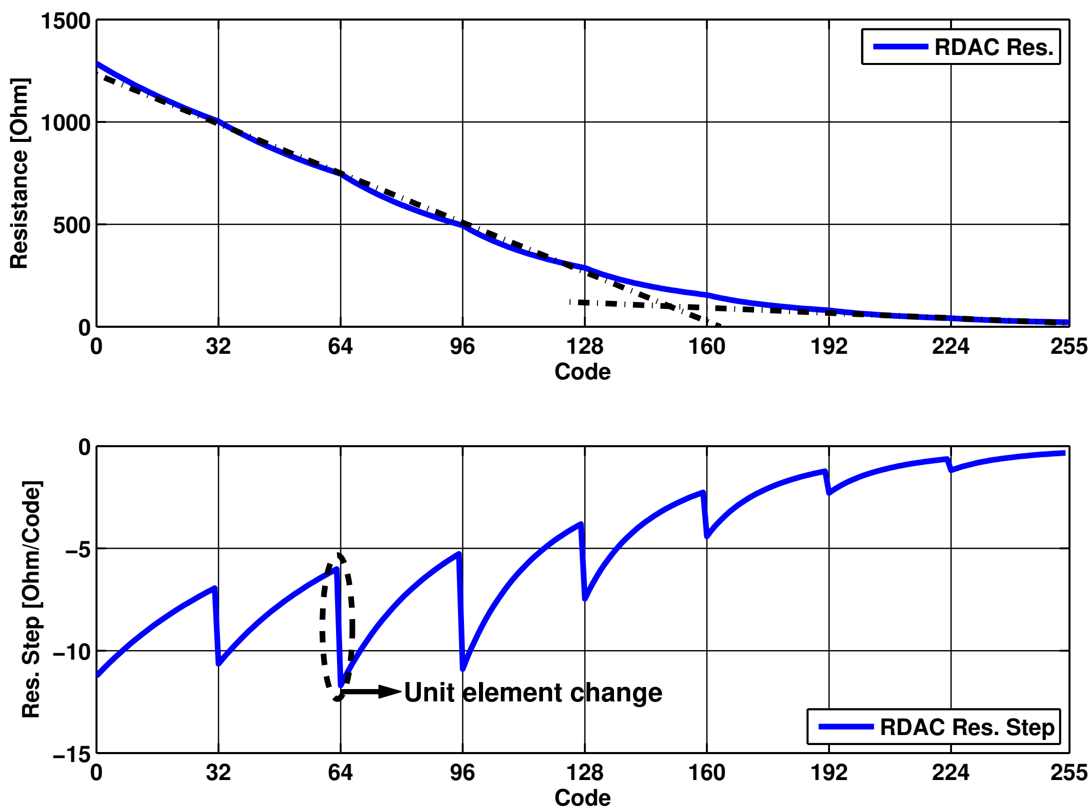


Figure 3.20: Simulated resistance and resistance step size of the P-RDAC as function of input digital code.

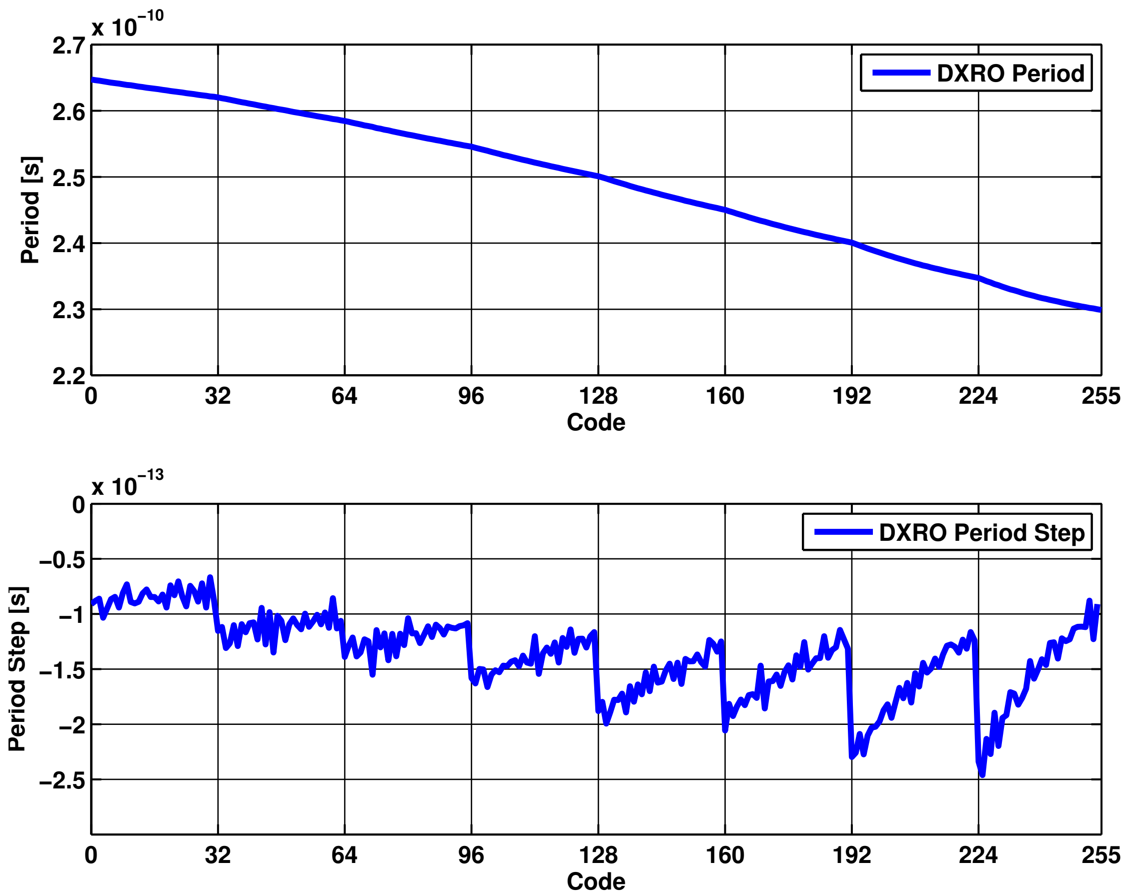


Figure 3.21: Simulated DXRO oscillation period and oscillation period step as function of input digital code.

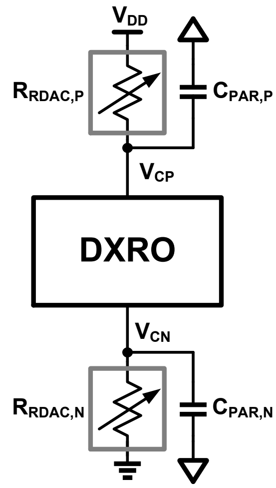


Figure 3.22: Illustration of poles associated with DXRO settling.

DXRO waiting time results in larger disturbance on V_{CP} and V_{CN} node voltages and further increases the settling time of V_{CP} and V_{CN}. On the other hand, faster initial settling on V_{CP} and V_{NP} results in smaller waiting time for DXRO and smaller disturbance on V_{CP} and V_{CN}. From this discussion, it can be concluded that for faster DXRO frequency settling, it is desirable to have

$$\tau_P, \tau_N < T_{REF} \tag{3.12}$$

From Fig. 3.23, it can be seen that both τ_P and τ_N satisfy the above condition under typical conditions. The slower settling of the V_{CP} node compared to the V_{CN} node can be attributed to larger $C_{PAR,P}$. Though layout routing parasitics are roughly the same on both V_{CP} and V_{CN} nodes, the larger size of PMOS devices connected to V_{CP} compared to NMOS devices connected to V_{CN} results in larger junction capacitance on V_{CP} node. It should be noted that $R_{RDAC,P}$ and $R_{RDAC,N}$ are maintained at roughly the same values to keep the output swing of the variable delay stages symmetric around $V_{DD}/2$.

3.4.2 Select Logic

The DXRO multiplexer select signal, SEL, has significant impact on MDLL performance. The timing of the SEL signal is critical in ensuring that the reference clock edge is propagated to the DXRO correctly. Sharp rise and fall times for the SEL signal are crucial to main-

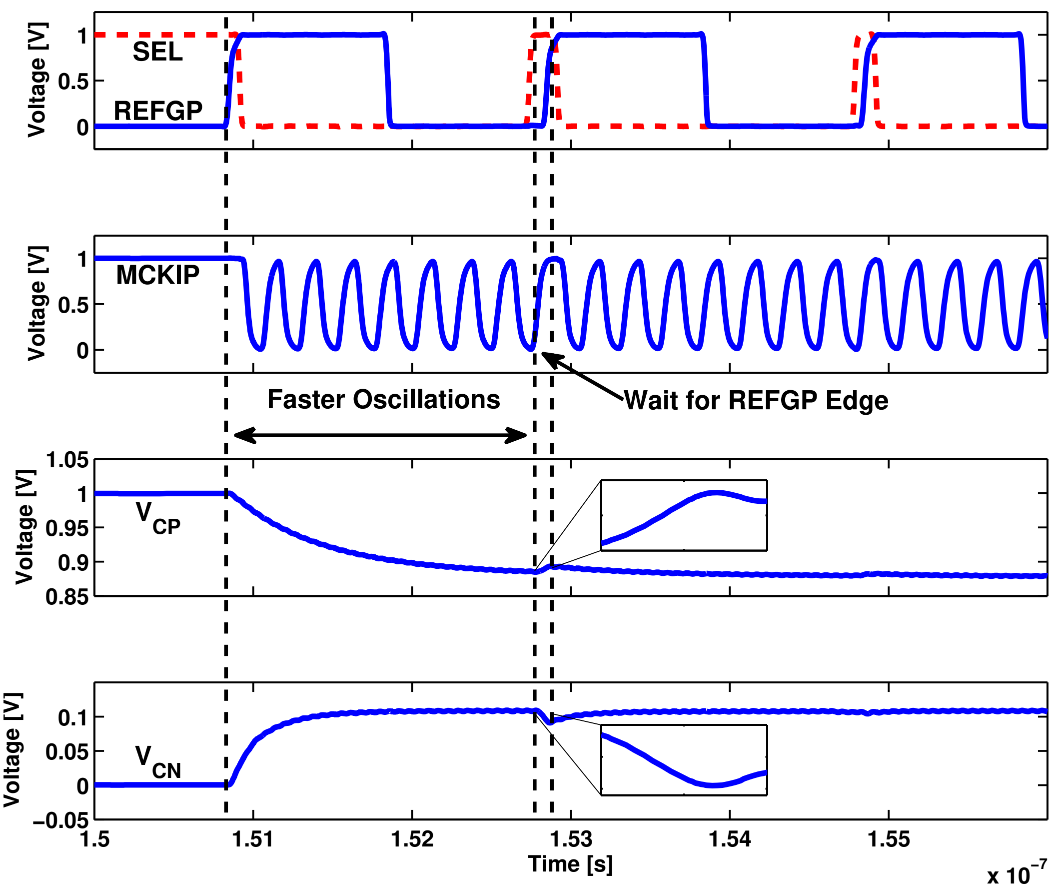


Figure 3.23: Simulated transient settling waveforms for DXRO turn-on.

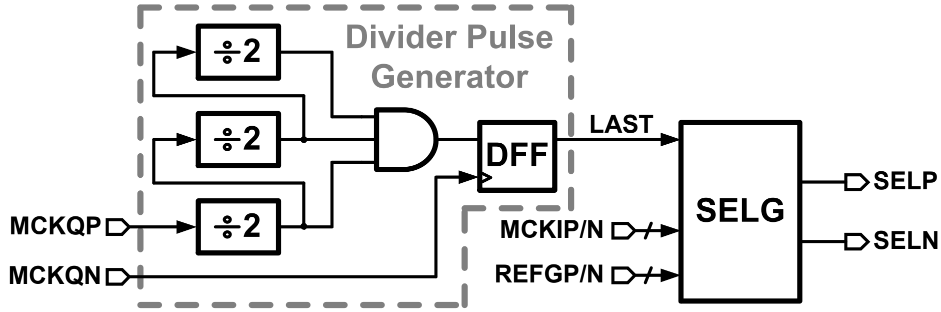


Figure 3.24: Select logic implementation block diagram.

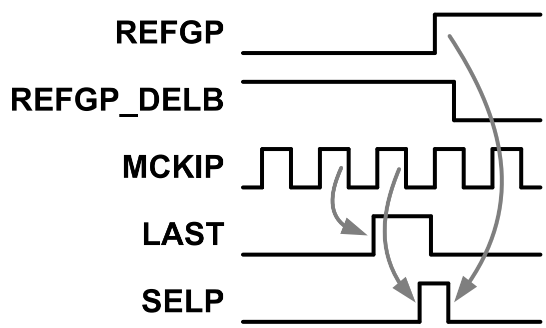


Figure 3.25: Select signal generation logic waveforms.

tain the reference clock waveform shape that is propagated to the DXRO. The select logic implementation is similar to [38] which consists of a divider and a select pulse generation block as shown in Fig. 3.24. Ripple counter configuration is used for the divide-by-8 block. Its outputs are then combined and retimed to generate a divider pulse every 8 DXRO clock cycles. Logic waveforms illustrating the operation of select signal generator block, SELG, are shown in Fig. 3.25. The select signal SELP is pulled to logic HIGH when the DXRO clock MCKIP goes to logic LOW if the divider pulse signal LAST is logic HIGH. SELP is pulled to logic LOW when both the reference clock REFGP and DXRO clock MCKIP go to logic HIGH. A delayed reference clock signal REFGP_DELB is used to improve frequency acquisition in case DXRO frequency falls below 8-times the reference frequency. It has no effect on MDLL performance during normal operation. To satisfy stringent timing requirements, the dynamic CMOS logic circuit scheme shown in Fig. 3.26 is used for generating the SELP / SELN signals. The back-to-back connected inverters ensure symmetric rise-fall time for both SELP and SELN signals. Additionally, the back-to-back connected inverters maintain the output logic state when MDLL is in off-state.

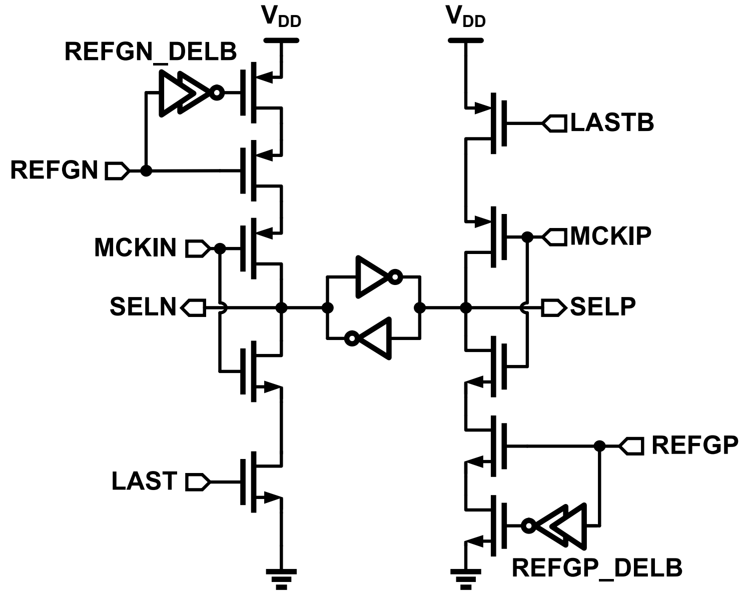


Figure 3.26: Select signal generation (SELG) circuit diagram.

3.5 Output Driver Design and Implementation

Figure 3.27 shows the top level block diagram of the output driver. A half rate architecture [44] with a 3-tap 2:1 serializer and a segmented CML output driver is used. In this work, use of CML output driver is preferred over a low swing voltage mode output driver. Use of low swing voltage mode output driver entails additional supply regulator based impedance control loop [10] for its output driver and pre-driver. This increases the complexity for rapid on-off operation. On the other hand, CML output driver utilizes passive terminations and does not need additional supply regulators for impedance control. The efficiency benefits of the voltage mode drivers are also diminished when additional pre-driver overhead and pre-emphasis is considered [21]. A ROOB circuit provides the bias voltage (V_B) for the CML output driver tail current sources. The ROOB circuit turns off the CML output driver when the transmitter is inactive and rapidly turns it back on when the transmitter is required to be active. The 3-tap 2:1 serializer multiplexes two 4 Gb/s bit-streams (D_{ODD}, D_{EVEN}) using the 4 GHz MDLL output clock MCK to generate pre-cursor (D_{-1}), main cursor (D_0), and post-cursor (D_1) data outputs at 8 Gb/s. The implementation details of CML output driver and ROOB block are described in the following sub-sections.

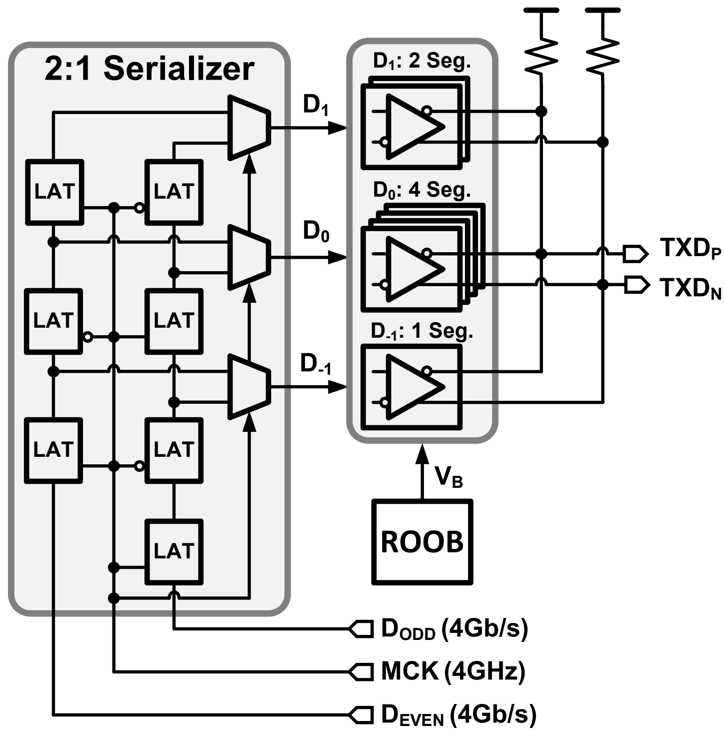


Figure 3.27: Top level block diagram of the 3-tap output driver.

3.5.1 CML Output Driver

The power dissipation of the CML output driver is split between the final output driver stage and the pre-driver stage. While the power consumption of the final stage can be scaled with output swing requirement, pre-driver stage power consumption cannot be easily scaled. Segmentation provides a way to coarsely scale pre-driver power consumption with output swing requirement by turning on only a required number of segments while the other segments are turned off. In this work, the main tap is connected to 4 segments while the pre-cursor and post-cursor taps are connected to 1 and 2 segments, respectively. The 2-bit tail current source in each segment can be used to set FFE coefficient and / or to control output swing of the output driver.

The circuit diagram of a unit segment is shown in Fig. 3.28. It consists of a CMOS to CML converter [45] that increases the common mode level of CMOS inputs coming from the 2:1 serializer and drives the CML pre-driver stage. To turn off the segment, the input to the CMOS to CML converter is gated such that both its inputs are at logic LOW level and ENB is pulled to logic HIGH level. This cuts off the current steering devices of the output driver

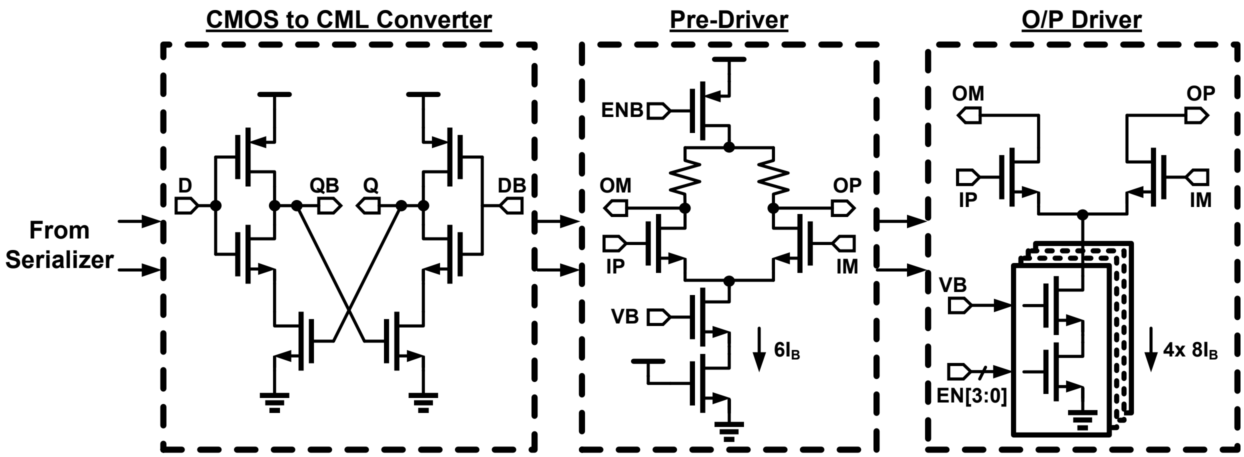


Figure 3.28: Unit segment of the CML output driver.

stage. The tail current sources of the output driver stage are also turned off by applying logic LOW input to EN[3:0].

3.5.2 2:1 Serializer

The 2:1 multiplexers used for serializing the data are very critical for the inter-symbol interference (ISI) performance of the transmitter. In the present implementation, two dynamic CMOS logic based 2:1 multiplexer circuits [15] are used in pseudo-differential manner to generate full-rate output as shown in Fig. 3.29. This multiplexer circuit provides good isolation between both even and odd data inputs as well as the output while avoiding excessive power consumption. To ensure appropriate logic levels for multiplexer inputs for avoiding floating nodes when the multiplexer is turned off, gating logic as depicted in Fig. 3.29 is implemented. Half rate complementary clock inputs CLK and CLKB are gated using an active low GATEB signal to generate gated clock outputs GCLKB and GCLK respectively. In addition to gating data signals (D_{ODD}, and D_{EVEN}) with GATEB signal, the data gating logic also uses SIGN input to apply appropriate polarity for the data signals required for 3-tap FFE. It should be noted that while gating logic maintains the complementary nature of its outputs during normal operation, all the outputs are pulled to logic high during off operation. This gating logic can be used not only during rapid on-off operation of the transmitter but also to selectively turn-off either one or both of pre-cursor and post-cursor taps

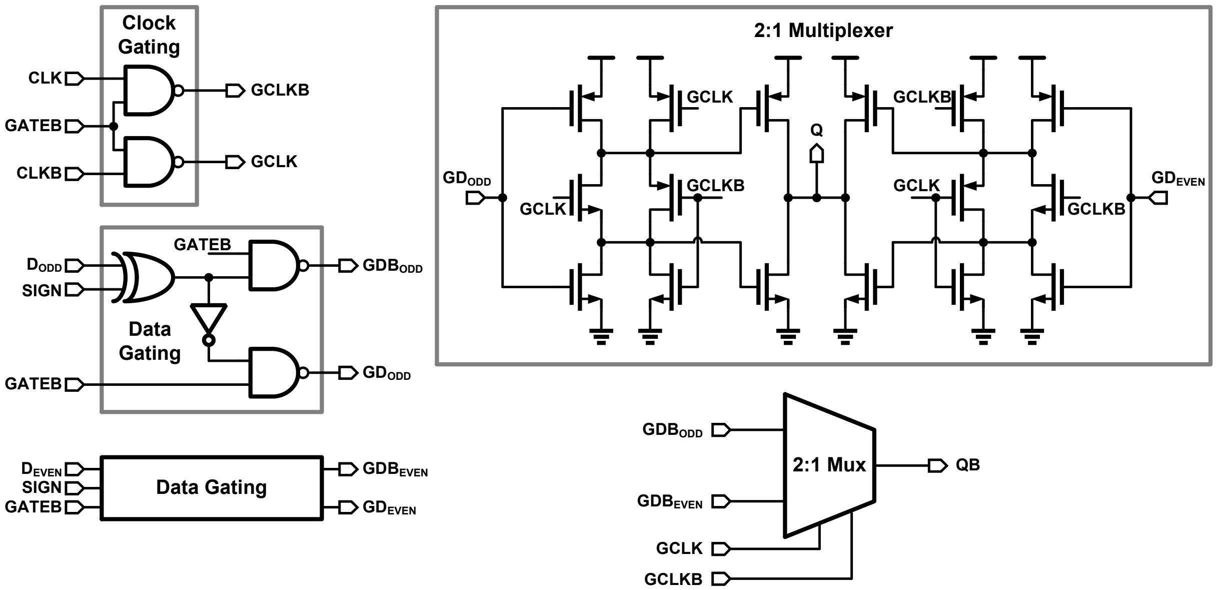


Figure 3.29: 2:1 serializer gating scheme and circuit diagram.

based on channel characteristic.

3.5.3 Rapid On-Off Biasing Scheme

The amplitude settling of the CML output driver mainly depends on its tail current source bias voltage settling during on-off operation. The conventional diode-connected MOS transistor based circuit used for current source biasing is shown in Fig. 3.30(a). If bias current I_B is cut off when transmitter is off, the turn on settling time of the bias voltage node VB is proportional to $C_B/g_{M_{NB}}$, where $g_{M_{NB}}$ is the transconductance of the diode-connected transistor M_{NB}. To achieve faster settling, we either need to increase the bias current I_B or decrease decoupling capacitance C_B. Increasing I_B results in an increased power consumption. On the other hand, C_B helps in mitigating effects such as coupling from other stray signals and ISI due to coupling between tail node voltage variations of CML stages and bias voltage node VB. Therefore reduction in C_B adversely affects output signal integrity.

To overcome this problem, we propose a calibrated ROOB circuit that provides an additional charging path for C_B to achieve faster settling. The proposed circuit occupies very small area and does not consume any static current. As illustrated in Fig. 3.30(b), the

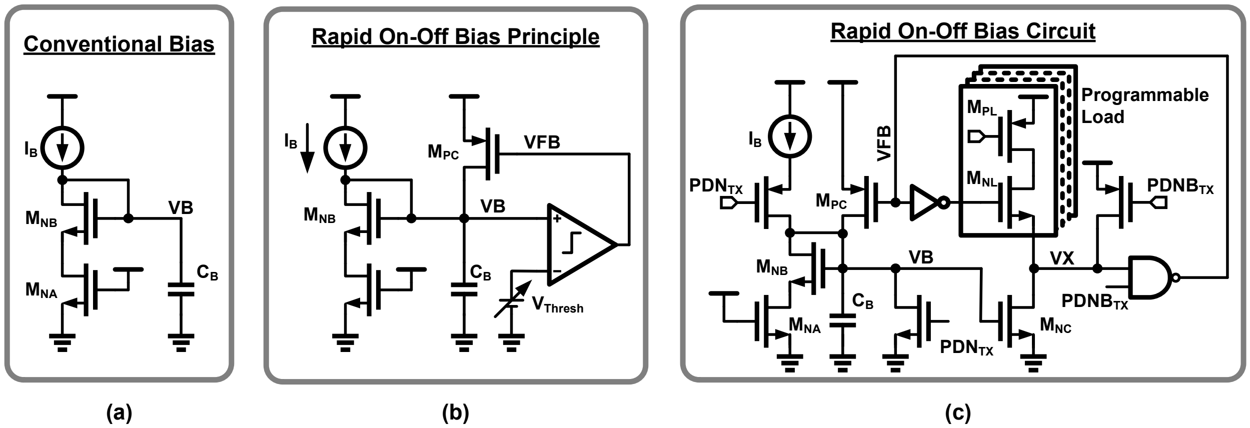


Figure 3.30: (a) Conventional diode connected current biasing scheme, (b) principle of operation for rapid on-off bias, and (c) the proposed rapid on-off biasing (ROOB) circuit.

basic principle behind operation of the ROOB circuit is to rapidly charge the capacitor C_B through M_{PC} until voltage of node VB reaches a value of V_{Thresh}. In the proposed circuit, V_{Thresh} is identified using simple digital calibration and the comparator is implemented using CMOS logic circuits. Referring to Fig. 3.30(c), the ROOB circuit operation can be described as follows. In off-state, $PDNB_{TX}$ and its complementary signal PDN_{TX} are logic LOW and logic HIGH, respectively. As a result, C_B is completely discharged (VB = 0) and node VX is pulled high. Consequently, the signal VFB is also pulled high. When ROOB is turned on by driving $PDNB_{TX}$ to logic HIGH, the diode connected transistor M_{NB} starts sinking current, and VB node voltage starts rising. After $PDNB_{TX}$ goes to logic HIGH, VFB goes to logic LOW, turning on the fast charging PMOS transistor M_{PC}. It helps VB node voltage to approach its steady state value rapidly, as indicated by the simulated waveform labeled VB_{ROOB} in Fig. 3.31. It can also be seen that using only diode connected transistor would result in much slower settling similar to the waveform labeled VB_{Dio} in Fig. 3.31. To avoid excessive overshoot on VB node voltage, a variable threshold inverter formed by a programmable load and M_{NC} is used. When node VB reaches the voltage where M_{NC} can overcome the pull up load, node VX is pulled to logic LOW. As a result VFB goes to logic HIGH and the additional charging path provided by M_{PC} is turned off. Subsequently, ROOB circuit behaves similar to a conventional diode-connected bias circuit as shown in the region marked "Bias Diode-like Settling" in Fig. 3.31. A 4-bit thermometer coded programmable load is used that provides around 10% initial settling accuracy after 4 ns with threshold calibration. The programmable load consists of a PMOS transistor M_{PL} and an NMOS transistor M_{NL}. The total resistance of the load is dominated by M_{NL}. M_{PL} is used to enable or disable the load branch. Use of NMOS transistor as main load provides better tracking of temperature variation compared to a PMOS only load.

During initial calibration of the programmable load, M_{PC} is disabled and the programmable load resistance is set to its lowest value. Beginning with completely discharged C_B, the diode connected transistor M_{NB} is allowed to slowly pull VB node to its steady state value. Due to stronger pull-up load, M_{NC} cannot pull the node VX down to ground. The state of VX node is detected by sampling signal VFB. The programmable load resistance is then increased and C_B is discharged. The diode connected transistor M_{NB} is again allowed to slowly pull VB node to its steady state value. This procedure is repeated until M_{NC} can pull the node VX down to ground. Corresponding code for load strength is used for subsequent rapid on-off operation of the transmitter. In current implementation, the initial calibration is performed manually but an automatic digital state machine based calibration can be easily implemented.

3.6 Effect of Voltage and Temperature Drift on ROO Operation

In typical electronic systems, supply voltage and temperature drift are the result of many factors such as long term stability of voltage references, ambient temperature, and on-chip power dissipation. For example, the study in [46] measured thermal time-constants, τ_{TH}, between 5 ms and 300 ms for die temperature change caused by variations in power dissipation. Such long time constants have little impact on the performance of burst-mode operation if the off durations (T_{OFF}) are relatively small ($T_{OFF} << \tau_{TH}$). For a very large T_{OFF}, it becomes necessary to intermittently turn on the interface to compensate for variations in operating conditions, albeit at a slightly reduced energy efficiency. Such system level techniques to reduce the impact of supply voltage and temperature drift on a burst-mode on-off interface are being actively explored [47]. In the following, we quantify the temperature and supply voltage dependence of the MDLL and ROOB.

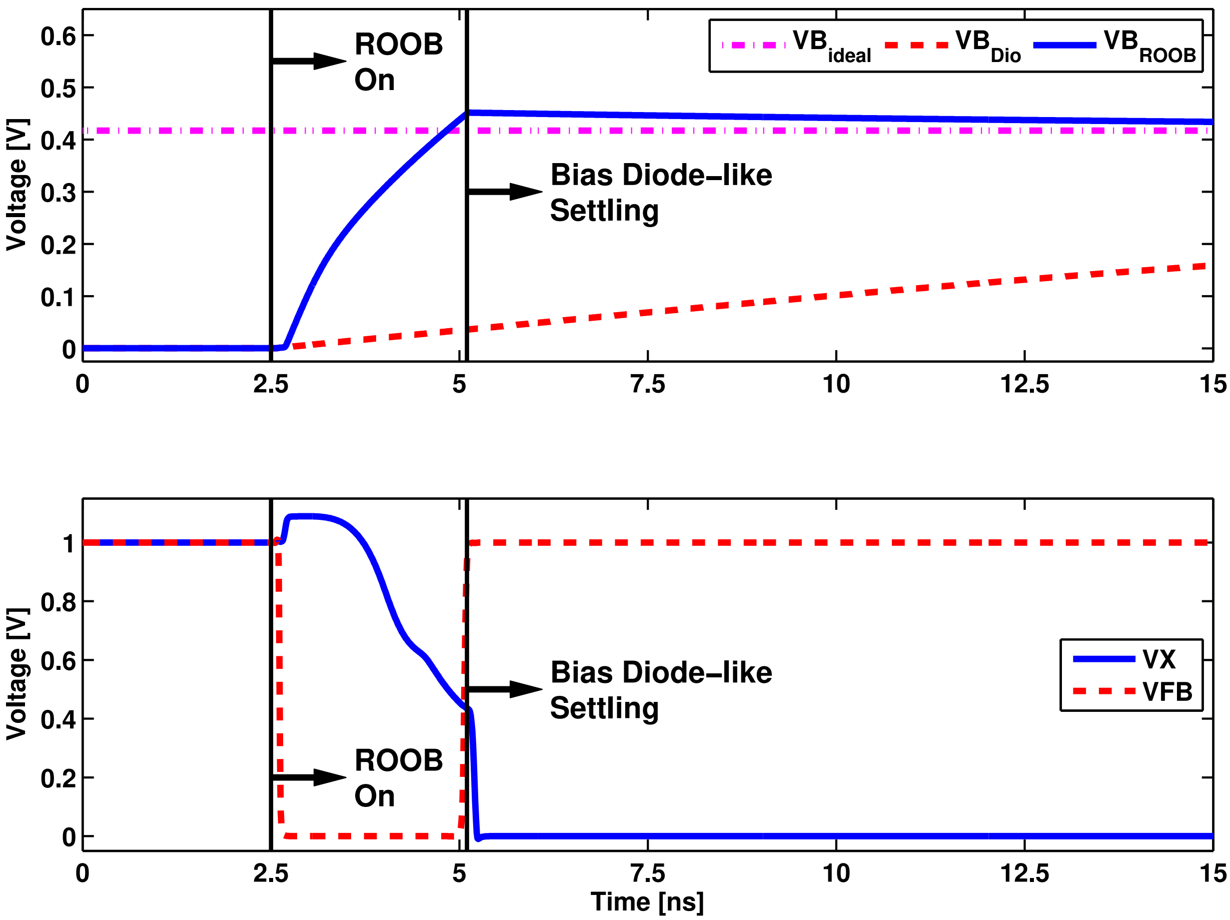


Figure 3.31: Simulated waveforms showing the ROOB settling process.

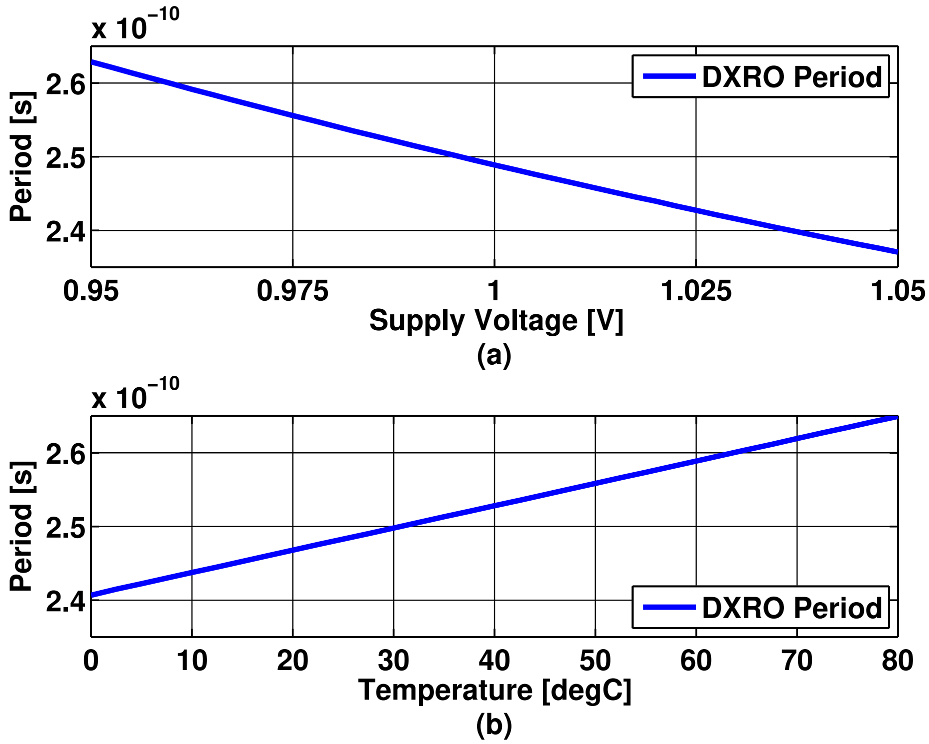


Figure 3.32: (a) Simulated DXRO time period as function of supply voltage, and (b) simulated DXRO time period as function of temperature.

3.6.1 Effect on MDLL

The on-off operation of MDLL results in disconnecting the MDLL feedback loop when the MDLL is in off-state. As a result the MDLL cannot track any changes in its operating conditions such as supply voltage or temperature that may occur during off-state. Under these conditions, MDLL sensitivity is essentially limited by open loop DXRO sensitivity. Figure 3.32(a) shows the simulated DXRO time period as function of supply voltage under typical conditions. Due to the use of MOS resistors for the DXRO frequency control, the present DXRO architecture is more susceptible to changes in supply voltage compared to a current controlled architecture (Fig. 3.15). From the figure, the DXRO supply sensitivity is found to be around $-258\,\text{fs/mV}$. A plot of DXRO time period as a function of temperature for a fixed input code is shown in Fig. 3.32(b). The simulated temperature sensitivity of the DXRO is around $303\,\text{fs}/^\circ\text{C}$.

It is important to note that the above values pertain to open loop DXRO. Once it is in on-state, the MDLL tries to track the changes in operating conditions during off-state. If the accumulator code does not represent the correct DXRO time period, the MDLL slews towards the correct code. The slewing of the accumulator code, and consequently the slewing of the DXRO period, is a function of the loop parameters as follows:

$$\text{MDLL Period Slew Rate[fs/ns]} = \text{Acc. Update Rate [LSB/ns]} \times \text{DXRO Gain [fs/LSB]} \tag{3.13}$$

For the present MDLL implementation, the slew rate is calculated to be

$$\begin{aligned} \text{MDLL Period Slew Rate[fs/ns]} &\approx \frac{1\,\text{LSB}}{128\,\text{ns}} \times \frac{-142\,\text{fs}}{1\,\text{LSB}} \\ &\approx -1.11\,[\text{fs/ns}] \end{aligned}$$

This information can be used to calculate the impact of supply and / or temperature drift on the MDLL performance. Assuming that the change in DXRO period due to supply / temperature drift during off-state is within MDLL acquisition range, the initial deterministic jitter is given by

$$\mathrm{T}_{\text{DJ,drift,p-p}} = 8\,\Delta\mathrm{T}_{\text{drift}} \tag{3.14}$$

where $\Delta\mathrm{T}_{\text{drift}}$ is the change in DXRO period due to drift in supply voltage / temperature. The jitter reduces as MDLL tries to acquire lock again. The lock acquisition time is approximately given by

$$\mathrm{T}_{\text{acq}} = \frac{\Delta\mathrm{T}_{\text{drift}}}{\text{MDLL Period Slew Rate}} \tag{3.15}$$

From the above equations, it is seen that large changes in supply voltage / temperature during off-state may lead to increase in MDLL turn-on time and worse performance.

3.6.2 Effect on Rapid On-Off Biasing Circuit

Figure 3.33(a) and Fig. 3.33(b) show the simulated variation in ROOB settling error with supply voltage and temperature variation respectively. For the simulation, the ROOB is calibrated under typical supply voltage and temperature conditions following the procedure

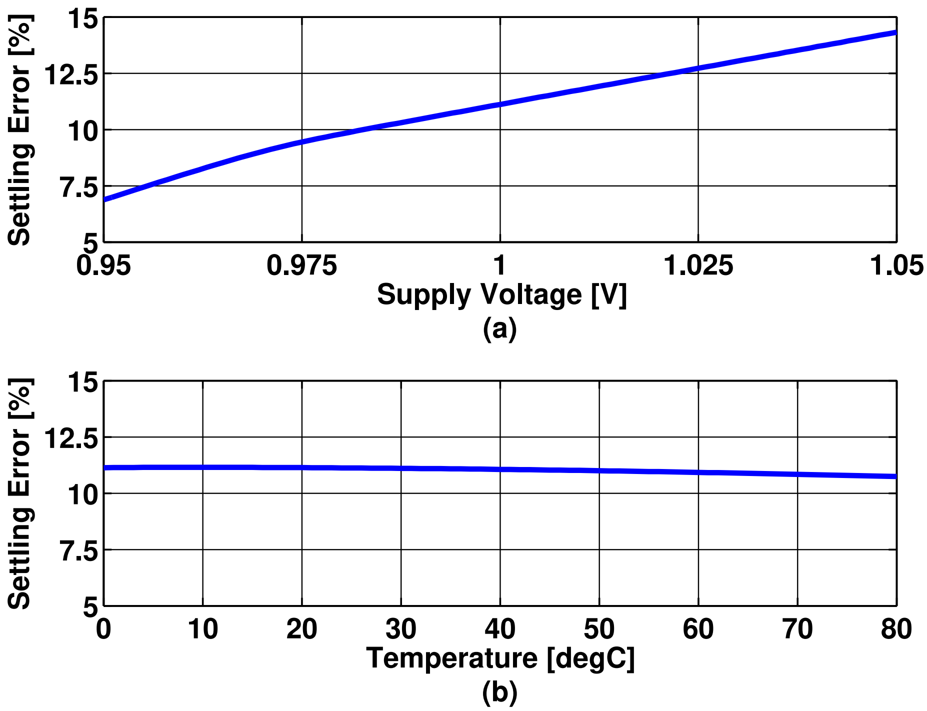


Figure 3.33: (a) Simulated ROOB settling error variation with supply voltage variation, and (b) simulated ROOB settling error variation with temperature variation, both evaluated at 4 ns after turn-on time.

outlined previously. The resulting code for programmable load strength is kept the same across supply voltage and temperature variations to measure resulting variation in settling error. The ROOB exhibits good tolerance to temperature variations due to use of NMOS load device. The sensitivity of ROOB to supply voltage can be attributed to change in threshold voltages of programmable load inverter as well as subsequent logic gates due to supply variation. As discussed in the case of MDLL, periodic calibration of ROOB circuit can be performed to track slow changes in supply voltage. The calibration process is simple and has very little additional circuit overhead.

3.7 Measurement Results

The prototype transmitter is implemented in 90 nm CMOS process and occupies 0.2 mm^2 active area. The die photograph is shown in Fig. 3.34. MDLL and the output driver each

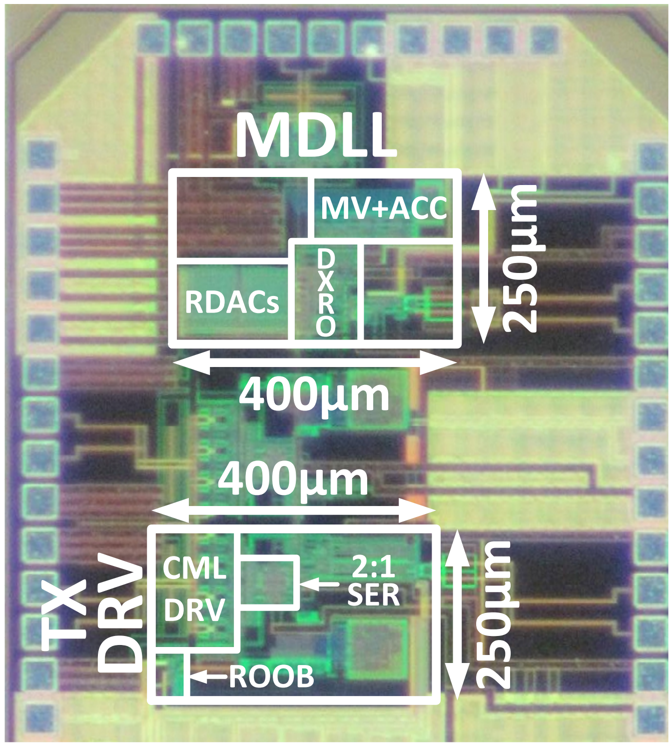


Figure 3.34: Die micrograph.

occupy 0.1 mm^2 area. Area of the output driver also includes a PRBS-7 pattern generator and a digital configuration block. We first describe the always-on transmitter measurements followed by rapid on-off measurements.

3.7.1 Always-On Measurements

The long-term jitter histogram with 1 million hits for the MDLL operating in always-on mode is shown in Fig. 3.35(a). The MDLL achieves an absolute jitter of 1.16 ps_{rms} and 13.2 $ps_{pk\text{-}pk}$ when the reference clock jitter is 0.8 ps_{rms}. The measured MDLL output phase noise plotted in Figure 3.35(b) shows that the integrated jitter is 300 fs_{rms} (10 kHz to 100 MHz). The MDLL output spectrum, shown in Fig. 3.36, shows a reference spur of -45.1 dBc in addition to multiple spurs occurring at offset frequencies that are harmonics of 62.5 MHz, caused by coupling between majority voting logic and DXRO. The largest such spur has magnitude of -44.3 dBc at 125 MHz offset frequency. The MDLL consumes 3.73 mW power at 4 GHz output frequency.

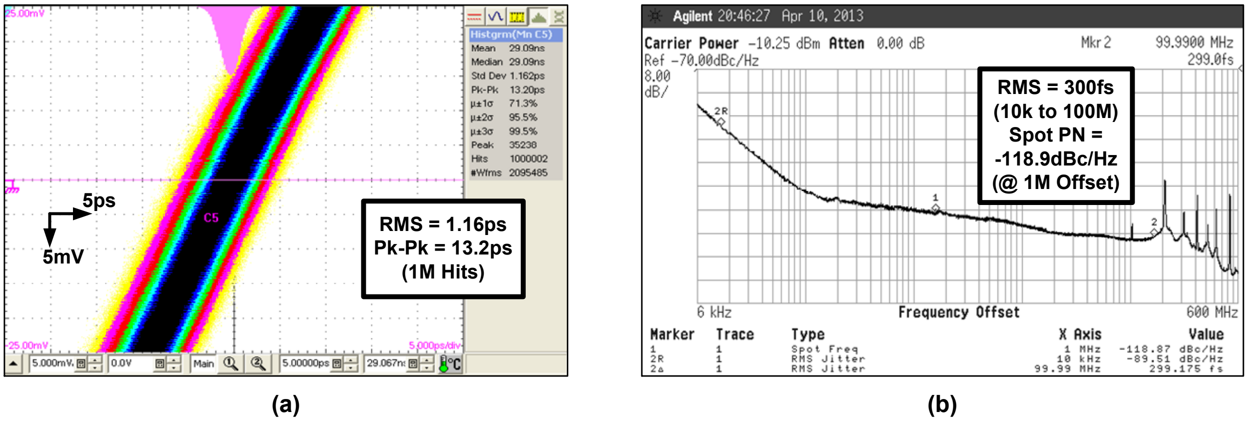


Figure 3.35: (a) Measured long-term time domain jitter of the MDLL, and (b) measured phase noise spectrum of the MDLL.

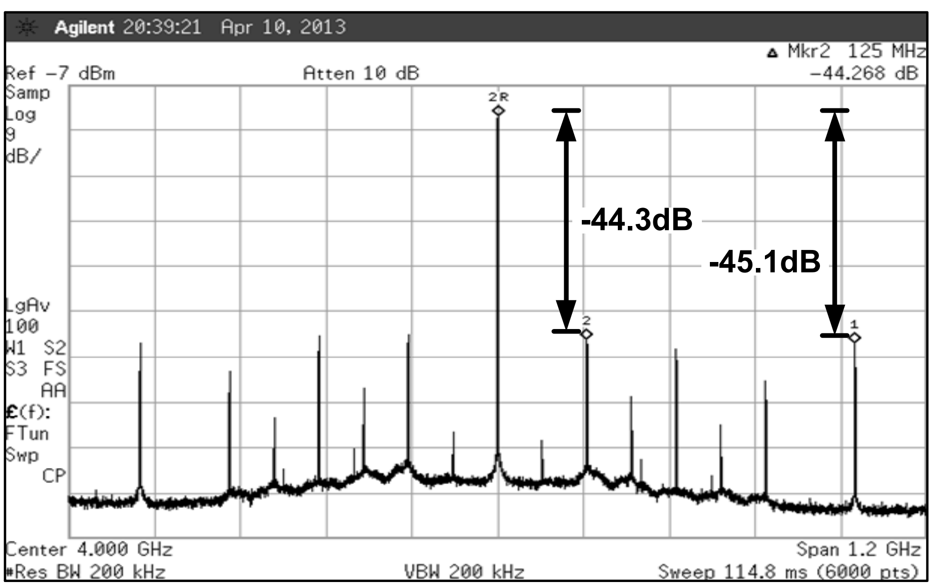


Figure 3.36: Measured output voltage spectrum of the MDLL.

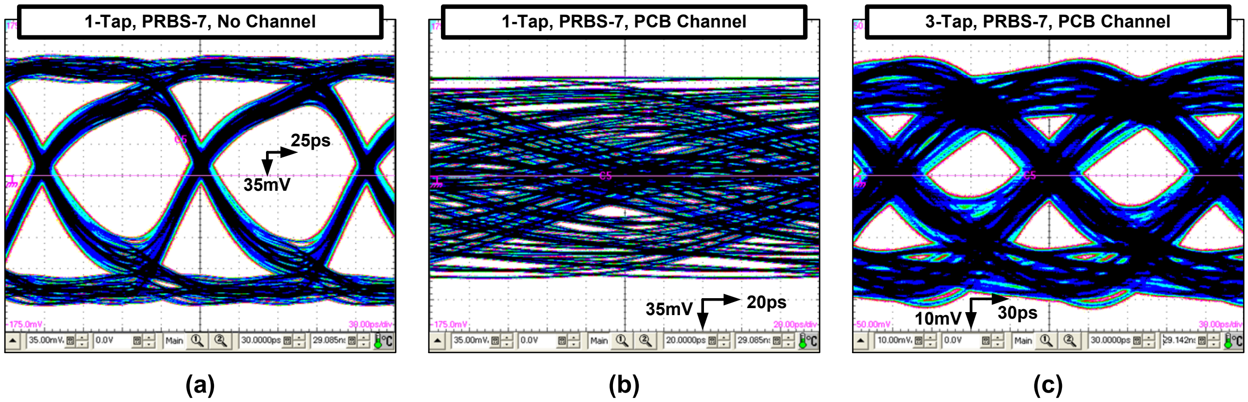


Figure 3.37: Transmitter data output eye diagram at 8 Gb/s data rate with always-on PRBS-7 output pattern for (a) 1-tap output with no channel, (b) 1-tap output with 13.4 dB loss channel, (c) 3-tap equalized output with 13.4 dB loss channel.

The measured transmitter output eye diagram in always-on condition at 8 Gb/s data rate is shown in Fig. 3.37(a). In 1-tap mode with PRBS-7 data, the transmitter jitter is 3.26 $\mathrm{ps_{rms}}$ and 23.2 $\mathrm{ps_{pk\text{-}pk}}$ after 100k hits. The loss due to package parasitics and PCB trace is estimated to be around 3 dB at 4 GHz. Figure 3.37(b) shows the unequalized transmitter output eye diagram at the end of a PCB channel that exhibits a loss of 13.4 dB at 4 GHz. Consequently, the output eye is almost closed. The equalized output eye diagram in 3-tap mode is shown in Fig. 3.37(c). The horizontal and vertical single ended output eye opening is 59 ps and 11.47 mV, respectively. The performance of the equalizer is limited by the duty cycle error amplification due to channel loss. The output driver, including the 2:1 serializer, consumes 14.56 mW power in 1-tap mode and 16.45 mW power in 3-tap mode for a 500 $\mathrm{mV_{ppd}}$ output swing.

3.7.2 On-Off Measurements

To test the on-off behavior of the MDLL, an arbitrary waveform generator (Tektronix AWG7122B) is used to generate the reference clock (REF) as well as the power down signal (PDNB). PDNB is also used to trigger the real time oscilloscope (Tektronix TDS6804B) and equivalent time oscilloscope (Tektronix DSA8200) waveform capture. This setup is shown in Fig. 3.38. Figure 3.39(a) depicts the measured MDLL output settling waveform captured using the equivalent time oscilloscope. The MDLL time period deviation during settling,

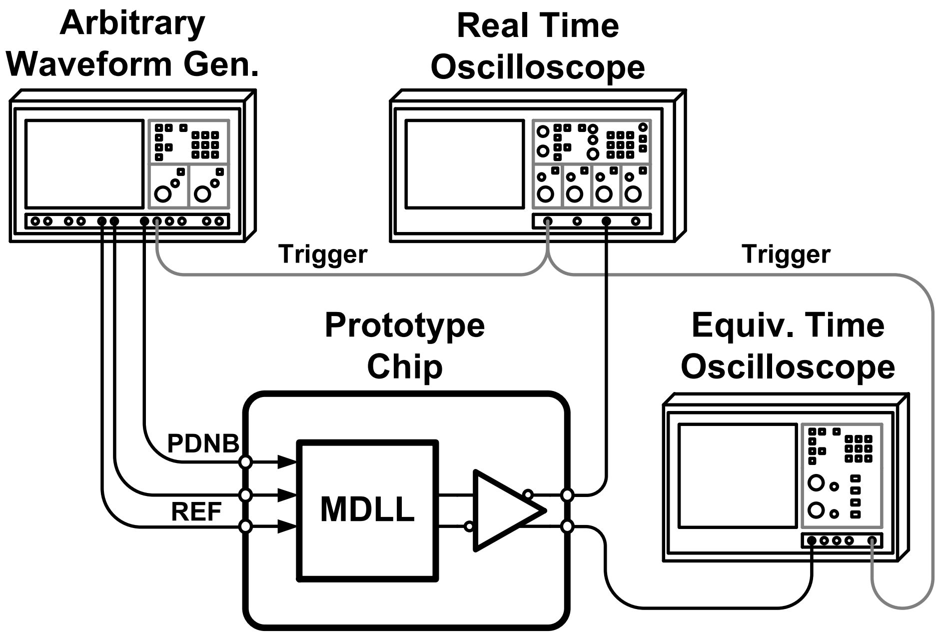


Figure 3.38: Measurement setup for on-off performance measurement of MDLL.

relative to its steady-state mean time period, is computed from the waveform data captured using real time oscilloscope and is shown in Fig. 3.39(b). The period error is always within $\pm 5\%$ during the settling transient, demonstrating the efficacy of the proposed DXRO architecture. After three reference cycles, period error is very close to the steady-state period jitter of the MDLL. Comparison of MDLL performance with the state of the art, shown in Table 3.1, illustrates that this work achieves superior jitter and power efficiency performance compared to conventional analog MDLL-based clock multipliers in addition to being amenable to rapid on-off operation.

The transmitter settling transient during turn-on, with and without the proposed ROOB circuit, is shown in Fig. 3.40(a). The measurements indicate that it takes more than 120 ns to settle without the ROOB circuit, and around 4 ns when the ROOB circuit is utilized (Fig. 3.40(b)), representing a nearly 30X improvement.

Figure 3.41 depicts the block-wise measured power consumption of the transmitter in always-on and off-state. The total on-state power is 18.29 mW (2.29 mW/Gb/s) whereas the off-state power is 0.11 mW in 1-tap mode. The off-state power is dominated by the sub-threshold leakage current in MDLL and serializer both of which use CMOS circuits and low-V_t transistors for high speed operation. The average power consumption for 4, 8, 32 and

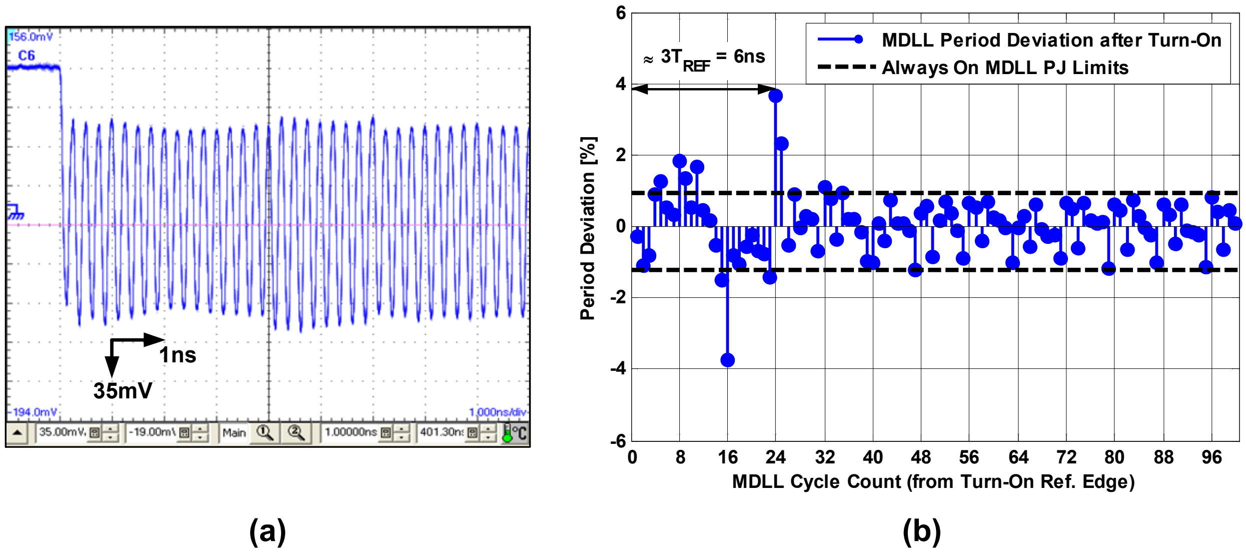


Figure 3.39: (a) MDLL output waveform during turn-on transient, and (b) extracted MDLL output period error during turn-on transient.

Table 3.1: Performance comparison with recently published MDLLs and rapid on-off clock multipliers.

	This Work	JSSC'10 [22]	VLSI'11 [23]	CICC'12 [36]	ISSCC'13 [48]	ISSCC'11 [49]
Technology	90 nm	40 nm	40 nm	65 nm	90 nm	90 nm
Supply [V]	0.95	1.1	–	1.1	1.1	1.2/1.0
Output Freq. [GHz]	4.0	4.3	2.8	3.16	2.5	4.6
Ref. Freq. [MHz]	500	537.5	700	1000	312.5	575
Long Term Jitter rms/pk-pk [ps]	1.16/13.2	– / –	–/11[a]	–/30[a]	2.0/18.6	1.99/17.8
Ref. Spur [dBc]	-45.1	–	–	–	–	-46
Power [mW]	3.73	–	45.3	96[b]	2.2	6.8
Power Eff. [mW/GHz]	0.93	–	16.18	30.38[b]	0.88	1.48
Rapid On-Off Functionality	Yes	Yes	Yes	Yes	Yes	No
Turn-on Time [ns] # Ref. Cycles	6 3	241.8 130	8 5.6	10 10	10 3	N/A N/A
Architecture	DMDLL	PLL	MILO	MILO	DMDLL	MDLL
Area [mm^2]	0.1	–	–	0.149[b]	0.16	0.025

[a] Deterministic jitter [b] Includes output drivers

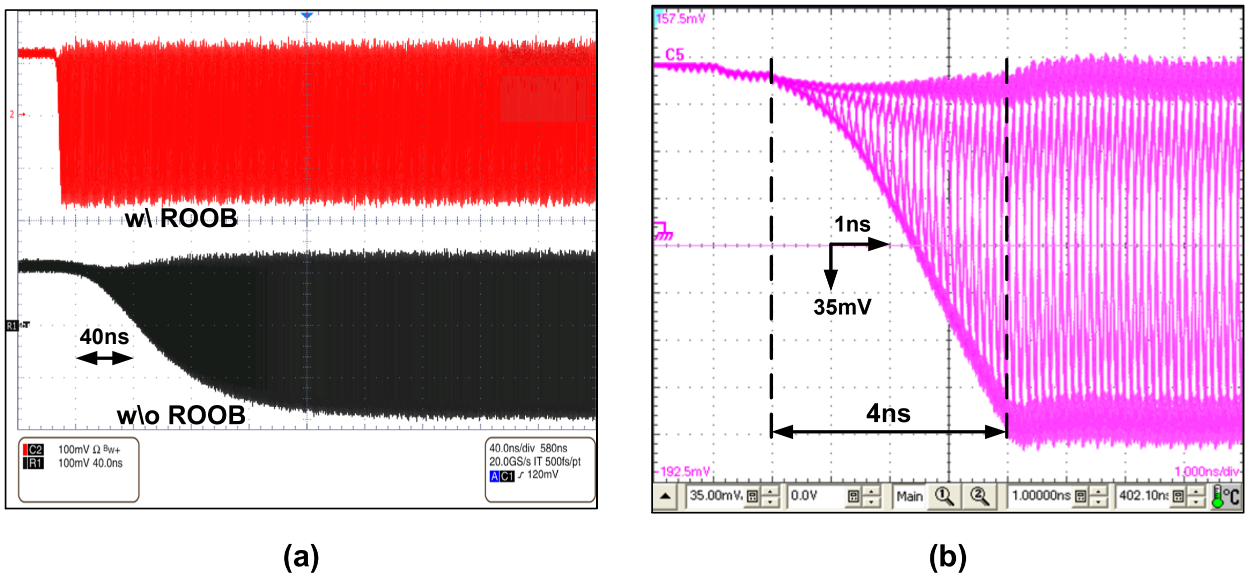


Figure 3.40: (a) Comparison of transmitter output swing settling with and without ROOB scheme, and (b) zoomed-in view of transmitter output swing settling with ROOB scheme during turn-on transient.

128 byte long bursts as a function of effective output data rate for 6 ns turn-on time is plotted in Fig. 3.42(a). All the measurements are made with PRBS-7 data output, 1-tap transmitter mode and without any additional PCB channel. As illustrated in Fig. 3.42(b), for each burst mode transfer, total on-time of the transmitter is decided by initial turn-on latency time and data transfer time. For example, to achieve 2 Gb/s effective data rate with 4 byte long burst, the transmitter is on for the first 10 ns, out of which 6 ns is turn-on latency time and data transfer time is 4 ns at the data rate of 8 Gb/s. The transmitter is turned off for the following 6 ns. Owing to the short turn-on time, the transmitter power consumption is proportional to effective data rate translating to almost constant energy efficiency. When the effective data rate is changed by 125X (8 Gb/s to 64 Mb/s), transmitter power consumption scales by 67X (18.29 mW to 0.27 mW) resulting in only 2X degradation in energy efficiency (2.29 mW/Gb/s to 4.24 mW/Gb/s) for 32-byte data bursts. At ultra-low effective data rates, the off-state power starts dominating and therefore energy efficiency degrades at a much faster rate. The transmitter transition energy is computed to be 95 pJ. Benefits of operating all the blocks in rapid on-off mode are evident from Table 3.2, which shows estimated energy efficiency that can be achieved by selectively operating one or more blocks in rapid on-off mode for

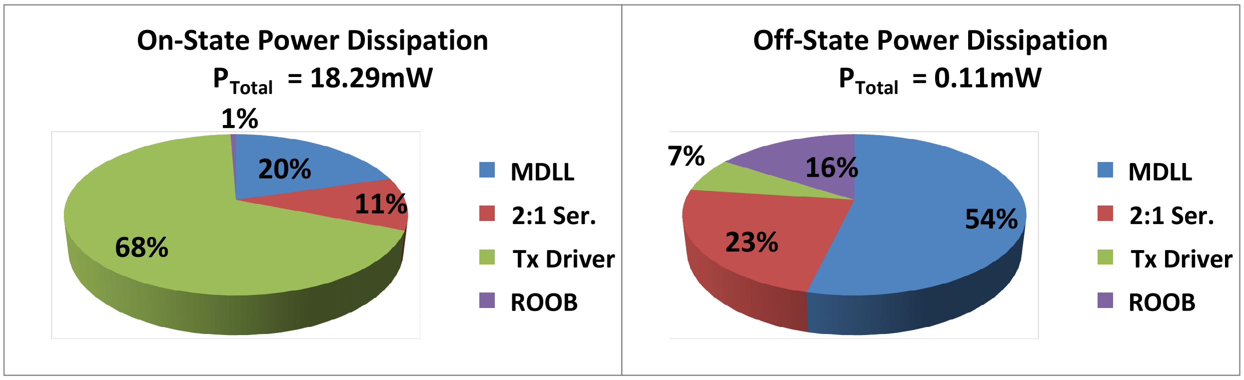


Figure 3.41: Power consumption of the transmitter for on- and off-states.

Table 3.2: Estimated energy efficiency for 32-byte transfer with average data rate of 64 Mb/s.

Mode	O/P Driver	MDLL	ROOB	Est. Energy Eff.
All Always-On	Always-On	Always-On	Always-On	285.7 pJ/b
P1	Rapid On-Off	Always-On	Always-On	62.7 pJ/b
P2	Rapid On-Off	Rapid On-Off	Always-On	5.9 pJ/b
All Rapid On-Off	Rapid On-Off	Rapid On-Off	Rapid On-Off	4.2 pJ/b[a]

[a] Measured energy efficiency

32-byte long burst-mode transfer at an average data rate of 64 Mb/s. Table 3.3 compares the transmitter with state-of-the-art CML output driver based transmitters. The energy efficiency of the transmitter in always-on state is comparable to other transmitters, while further energy savings can be obtained by rapid on-off operation of the proposed transmitter.

3.8 Analytical On-Off BER Computation

While the MDLL period error settling characteristic indirectly indicates how quickly an on-off interface can return to its steady state, it does not provide information about the settling time after which the I/O interface can be used to obtain the required bit error rate (BER) performance. Based on information obtained from measurements, it is possible to statistically predict the effect of MDLL settling characteristic on the transmitter performance. To simplify the analysis, we assume the overall link architecture shown in Fig. 3.43. We also

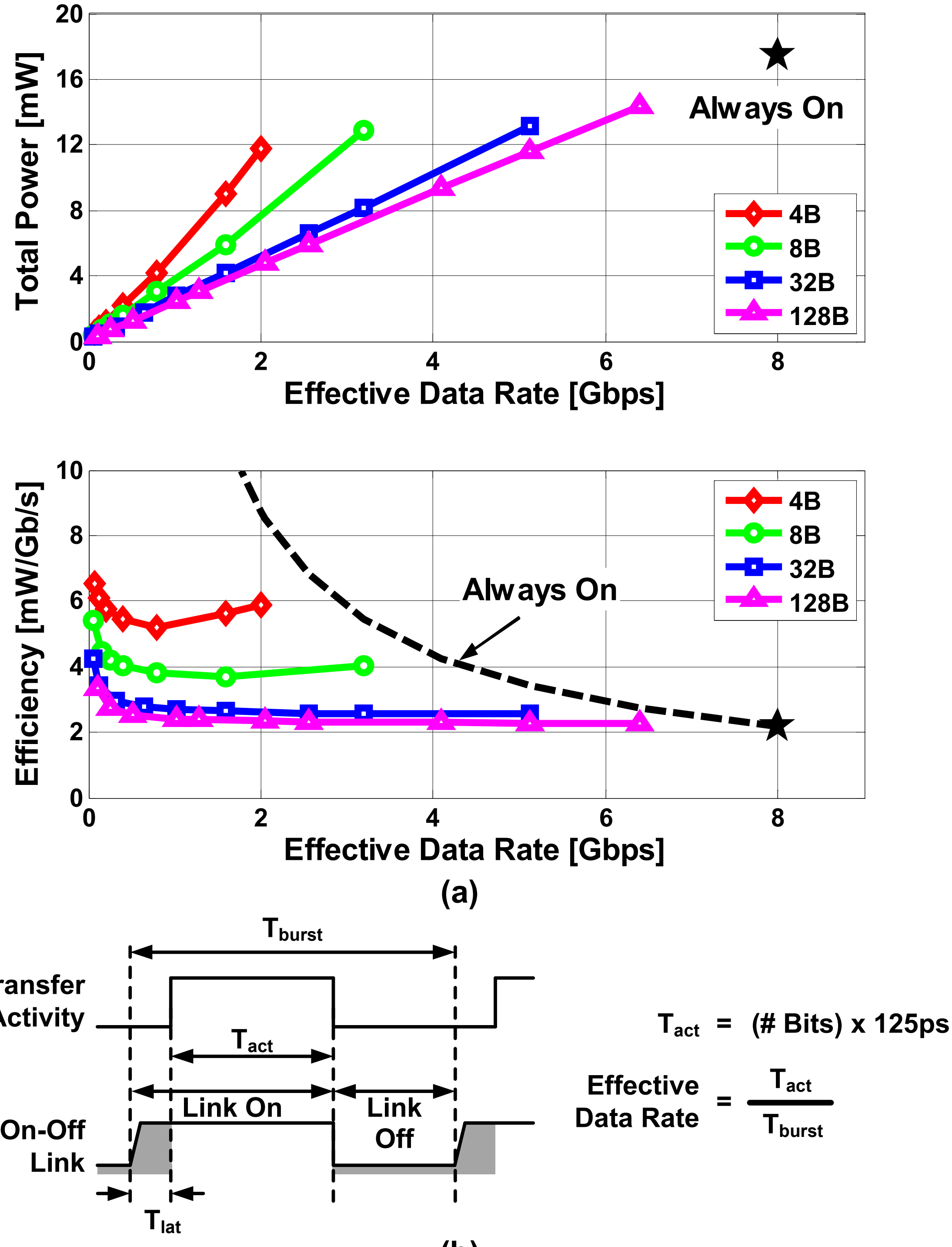


Figure 3.42: (a) Power consumption and energy efficiency of the transmitter for varying effective data rates, and (b) illustration of data transfer pattern used for varying effective data rate in measurement.

Table 3.3: Transmitter performance comparison.

	This Work	VLSI'11 [23]	JSSC'08 [21]	JSSC'10 [11]
Technology	90 nm	40 nm	65 nm	45 nm
Supply [V]	0.95/1.0	–	0.85/1.2	0.80
Output Data rate [Gb/s]	8.0	5.6	10.0	10.0
O/P Swing[mV_{ppd}]	500	–	100	150
Tx Equalization	3-Tap FIR	None	3-Tap FIR + Pass. RL	2-Tap FIR
Tx Power [mW]	14.56[a] / 16.45[b]	–	17	5.28
Tx Eff. [mW/Gb/s]	1.82[a] / 2.06[b]	–	1.70	0.53
Rapid On-Off Bias (Settling Time)	Yes ($\sim$4 ns)	Yes (<2 ns)	No	No
Overall Eff. [mW/Gb/s]	2.29[a,c] / 2.52[b,c]	2.4[d]	3.6[e]	1.40[f]

[a] 1-Tap mode [b] 3-Tap mode [c] Includes 1 Tx lane with 1 MDLL
[d] Extrapolated for 8 Tx+Rx links [e] Includes 1 lane Tx+Rx
[f] Includes 47 lane Tx+Rx, 1 lane fwd. clock, 1 IL-VCO

make the following assumptions:

- The on-off data transmission does not affect the receiver sampling phase which is nominally at the center of the data eye in steady-state.
- The effect of ROOB settling is ignored and the output driver swing does not change during on-off operation.
- Data is transferred in bursts consisting of a fixed number of bits N_B.
- Data dependent jitter (DDJ) probability distribution function (pdf) remains the same.

Under these assumptions, the effect of MDLL period settling due to on-off operation on receiver sampling is shown in Fig. 3.44. Due to initial frequency settling of the MDLL, the receiver cannot sample the data in the middle of the data bit. If the MDLL zero crossing time instants ($t[i]$) are known, time instants corresponding to the middle of each data bit can be calculated. We denote the difference between the middle of the data bit and the corresponding receiver clock zero crossing time instant by $\varepsilon_s[i]$. The sequence $\varepsilon_s[i]$ represents

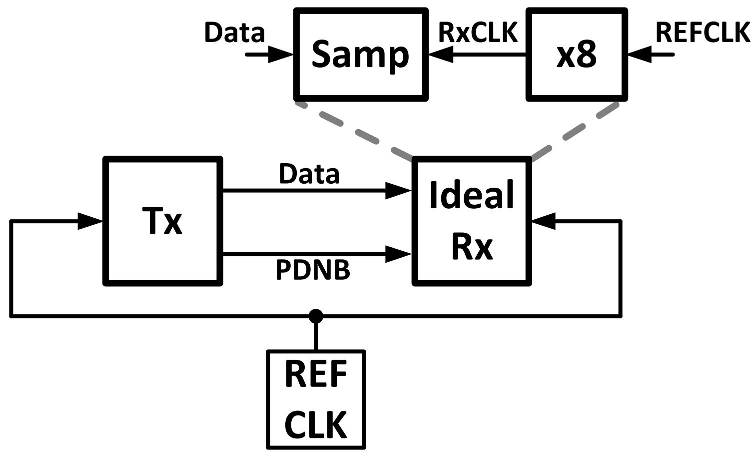


Figure 3.43: Link configuration used to analyze effect of on-off operation on link BER.

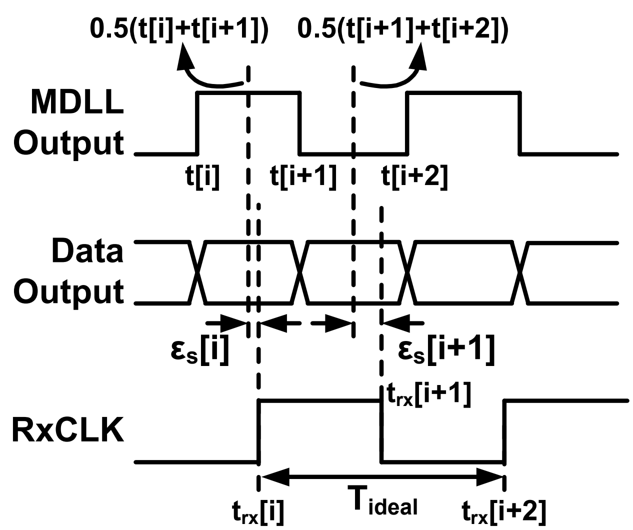


Figure 3.44: Illustrated sampling errors induced by MDLL frequency settling with ideal receiver clock.

the receiver sampling error due to imperfect frequency settling of the MDLL. Further, if we also know the steady-state total jitter pdf of the data at the receiver, we can construct a statistical BER bathtub as follows.

- For interface turn-on time $\mathrm{T}_{\mathrm{lat,on}}$, compute the receiver sampling error sequence $\varepsilon_s[i]$.
- Let the random variable corresponding to total jitter on data be given by $E_T[i]$ for ith zero crossing. Also assume that data values $d[i]$ are independent identical random (i.i.d.) variables.
- In such case, probability of error at a phase offset ϕ, normalized to 1 UI, for ith bit (b_i) is given by

$$\begin{aligned} P_i(\phi) &= \mathbb{P}\{\varepsilon_s[i] - E_T[i] + \phi < -0.5\,\mathrm{UI}\} + \mathbb{P}\{\varepsilon_s[i] + E_T[i+1] + \phi > 0.5\,\mathrm{UI}\} \\ &= \mathbb{P}\{E_T[i] > 0.5 + \varepsilon_s[i] - \phi\} + \mathbb{P}\{E_T[i+1] > 0.5 - \varepsilon_s[i] - \phi\} \end{aligned} \tag{3.16}$$

  where $\mathbb{P}\{\bullet\}$ denotes probability of an event.

- Let $F_{E_T}(\phi)$ be the cumulative probability distribution function of the i.i.d. random variable $E_T[i]$ at a phase offset ϕ.
- Then probability of error at the ith bit (b_i) can be written as

$$P_i(\phi) = 1 - F_{E_T}(0.5 - \varepsilon_s[i] - \phi) + F_{E_T}(-0.5 - \varepsilon_s[i] - \phi) \tag{3.17}$$

- The overall average probability of error for $\mathrm{N_B}$ bit burst is given by

$$\begin{aligned} P_{\mathrm{avg,\,N_B}}(\phi) &= \frac{1}{\mathrm{N_B}} \sum_{i=1}^{\mathrm{N_B}} P_i(\phi) \\ &= \frac{1}{\mathrm{N_B}} \sum_{i=1}^{\mathrm{N_B}} \left(1 - F_{E_T}(0.5 - \varepsilon_s[i] - \phi) + F_{E_T}(-0.5 - \varepsilon_s[i] - \phi)\right) \end{aligned} \tag{3.18}$$

The above expression can be used to compute the BER bathtub characteristic. The 80SJNB software tool [50] in conjunction with Tektronix DSA8200 is used to obtain $F_{E_T}(\phi)$

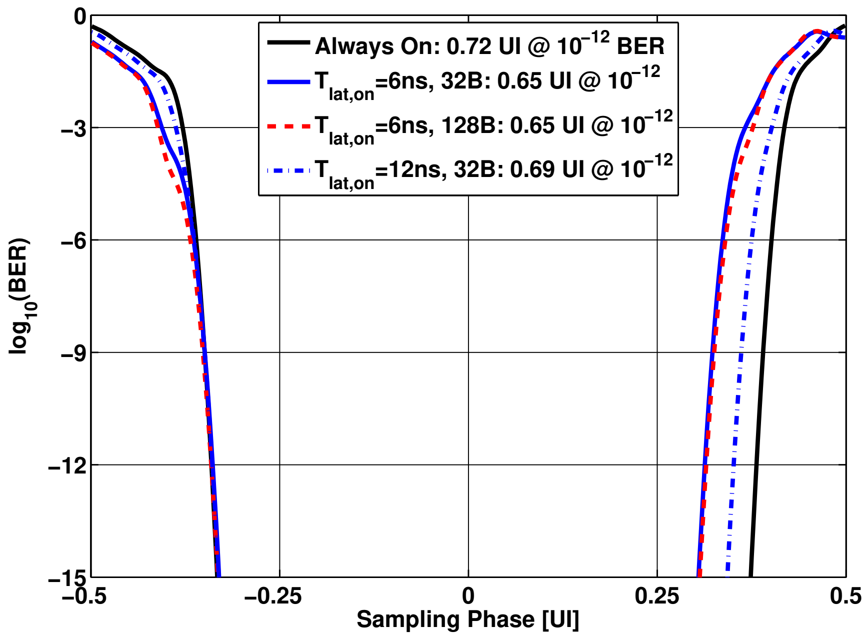


Figure 3.45: Always-on and on-off BER bathtub curves obtained using MDLL on-off settling measurements and statistical analysis.

under always-on condition for PRBS-7 data output at 8 Gb/s. MDLL period settling characteristic shown in Fig. 3.39(b) is used to compute the BER bathtub characteristics for varying burst lengths and turn-on latency, $T_{lat,on}$.

Figure 3.45 shows such computed BER bathtub characteristics for 32-byte, and 128-byte long bursts for $T_{lat,on}$ of 6 ns, for a 32-byte burst with $T_{lat,on}$ of 12 ns, and the always-on BER bathtub characteristic. It can be seen that the BER bathtub width for 10^{-12} BER is strongly affected by $T_{lat,on}$ whereas burst length, N_B, plays only a minor role. The BER bathtub width marginally improves with increasing N_B. To further illustrate the effect of $T_{lat,on}$ on width of BER bathtub, a plot of BER bathtub width as a function of $T_{lat,on}$ for 32-byte long transfers is shown in Fig. 3.46. Further analysis suggests that the transmitter BER settling is mostly unaffected in the presence of longer PRBS sequence length or higher channel loss. This is because initial timing errors are mainly dominated by MDLL frequency settling rather than data dependent jitter (DDJ). It should be noted that the steady-state BER bathtub width will be smaller with larger DDJ, similar to conventional always-on transmitters.

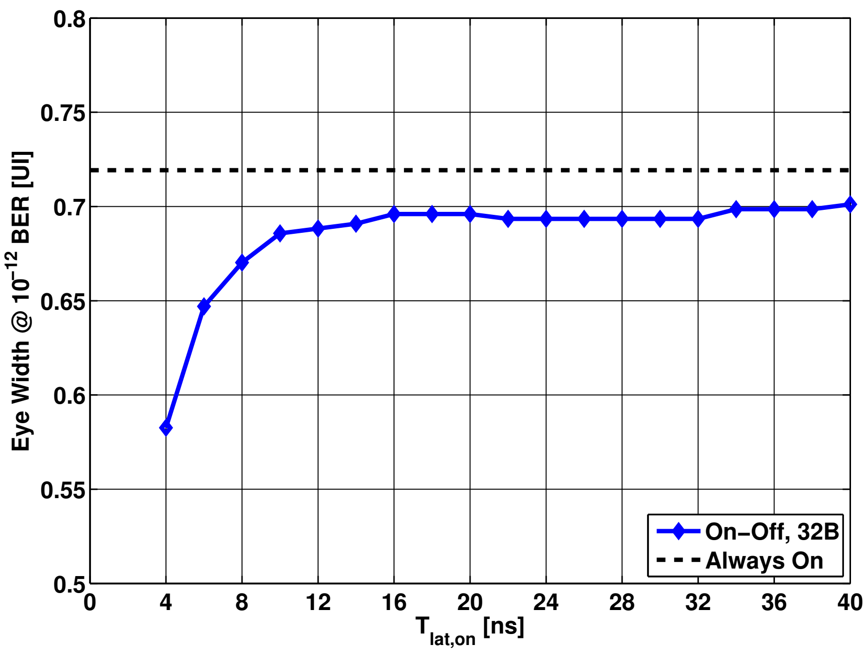


Figure 3.46: Width of BER bathtub curve as function of $T_{\mathrm{lat,on}}$ for 32-byte long burst mode transfer.

CHAPTER 4

DESIGN OF A BURST-MODE RECEIVER

After discussing the design of a burst-mode transmitter in Chapter 3, we turn our attention to the design of a burst-mode receiver in this chapter. As data rates for wireline links continue to rise, signal integrity has become worse for channel of fixed length due to increased losses at high frequencies. Therefore, the receivers that operate over these channels need to compensate for higher channel losses. We explore burst-mode phase locking techniques that can be utilized by receivers operating over high loss channels. Such receivers typically use a combination of equalizer structures such as continuous time linear equalizer (CTLE), discrete time feed-forward equalizer (FFE), and decision feedback equalizer (DFE) [51, 44, 52, 53, 54, 55, 56]. Furthermore, analog-to-digital converter-based architectures are also being explored for receivers [57, 58, 59, 60, 61] operating over high loss channels.

There are two main types of clock recovery schemes that are used in receivers for high loss channels. In the first scheme, shown in Fig. 4.1, an auxiliary clock and data recovery (CDR) path that uses bang-bang phase detector based logic for clock recovery is used [51, 62]. The output of CTLE is sampled to generate data (D_D) and edge (D_E) samples for the CDR logic. The output of CDR logic controls the phase of sampling clocks by means of phase rotators. The equalizer path consisting of FFE and DFE provides the actual data output, D_{OUT}. Typically, separate phase rotators are used for equalizer path and CDR path to account for any delay mismatches in these two paths. The use of the auxiliary CDR path is necessitated by discrete time equalizer structure that cannot be used for conventional bang-bang phase detector-based clock recovery scheme. The separate CDR path also decouples equalizer coefficient adaptation from clock recovery. A baud rate CDR architecture [63, 55], shown in Fig. 4.2, does not use an auxiliary CDR path. Instead, it makes use of an additional set of samplers with thresholds denoted as V_{REFP} and V_{REFM}. Their outputs, D_{EUP} and D_{EDN},

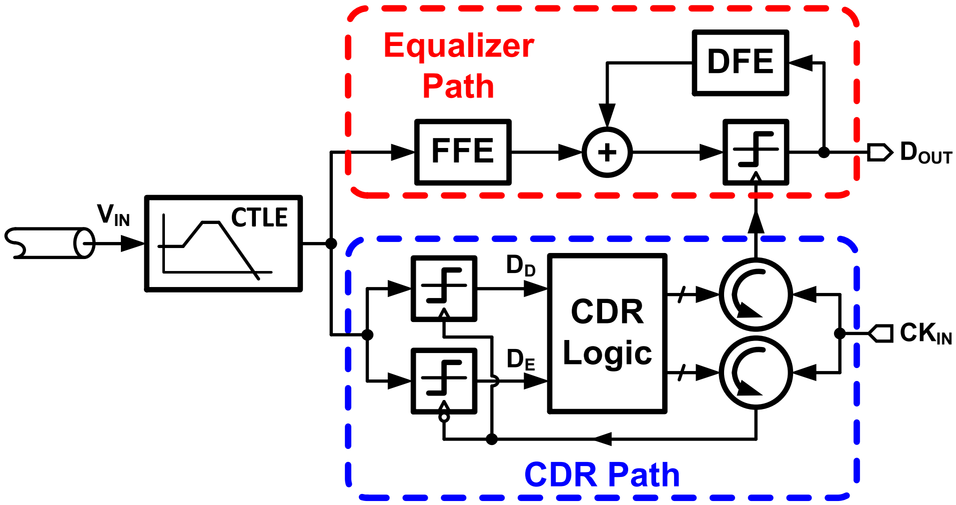


Figure 4.1: Receiver architecture with an auxiliary CDR path.

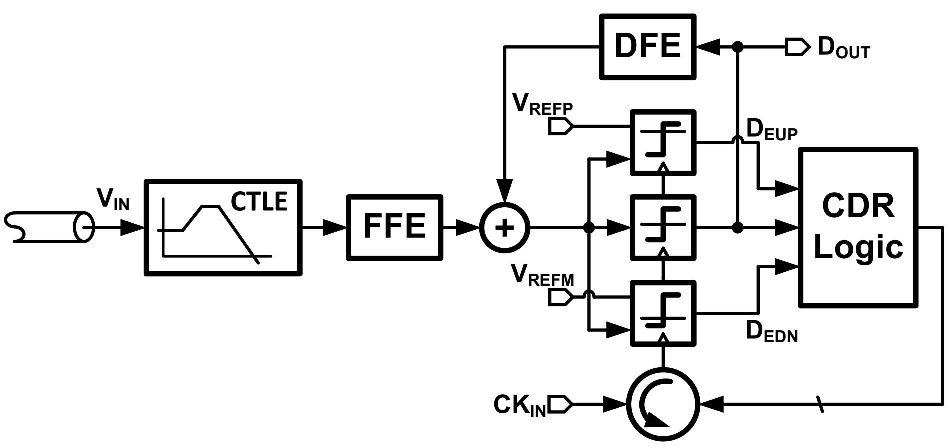


Figure 4.2: Baud rate receiver architecture.

along with the output of the main sampler, D_{OUT}, are processed by the CDR logic block to generate appropriate control input for the phase rotator. Typically, baud rate CDRs use the method described in [64] for phase detection. Baud rate CDR requires only one clock phase per bit period unlike edge-based CDR in Fig. 4.1 that requires two clock phases per bit period. To generate reference levels, V_{REFP} and V_{REFM}, baud rate CDR generally utilizes a DAC. It also necessitates use of an amplitude control mechanism in the analog front-end of the CDR. The ADC-based receivers commonly use the baud rate CDR architecture due to readily available additional samplers as well as reference signals [57].

For burst-mode operation of the receiver, the main roadblock is the fast synchronization of the CDR. This problem is similar to the fast-locking problem associated with clock multipliers discussed in Chapter 3, except that it is further exacerbated by the random nature of the data input. In contrast, the reference clock input to the clock multiplier has a deterministic pattern. The feedback-based operation of the CDR has a bandwidth limitation. This limitation mainly comes from the feedback latency of the CDR loop [55, 65]. Due to power constraints, the CDR processes the sampler outputs at a much lower frequency, resulting in increased latency. The input data also has large jitter, particularly when operating over high loss channels. To filter this jitter various techniques such as pattern filtering [66, 62] are used. These techniques invariably limit the CDR bandwidth.

To discuss the issues associated with burst-mode receiver design and to describe the proposed solution, this chapter is organized as follows. We begin with a brief overview of the prior art in Section 4.1. The principle of operation of the proposed fast synchronization method is described in Section 4.2. Section 4.3 provides the implementation details of the receiver chip that demonstrates the operation of the proposed fast synchronization method. Section 4.4 concludes this chapter.

4.1 Prior Art

While burst-mode receivers are necessary to realize I/O link architectures with "energy-proportional" behavior, they are more commonly used in passive optical networks (PON). Consequently, most of the prior research effort on burst-mode receivers has been focused on

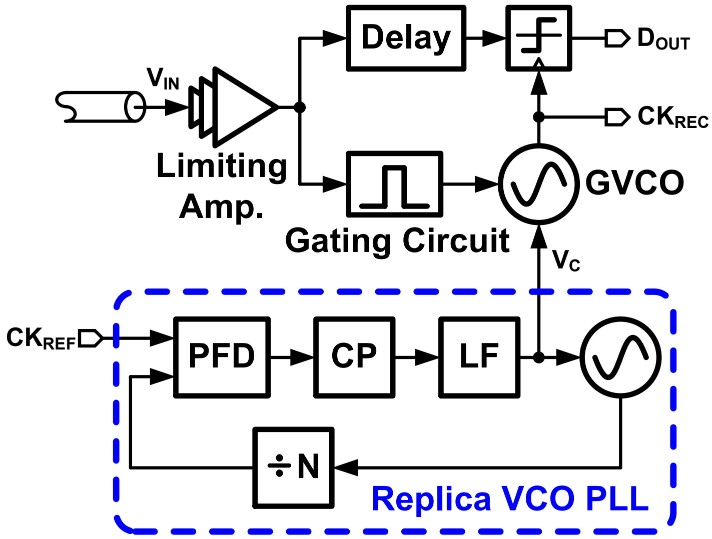


Figure 4.3: Gated VCO-based burst mode receiver architecture.

PON applications [67]. Use of gated voltage controlled oscillators (GVCO) is very common in burst mode receivers [68, 69, 70, 71, 72]. Figure 4.3 shows a simplified block diagram of a conventional GVCO-based burst mode receiver [70]. The receiver input V_{IN} is passed through a limiting amplifier to increase amplitude. A gating circuit generates short pulses for every data transition that can be injected into GVCO. Such injection aligns the phase of GVCO to input phase. The input is delayed and sampled using recovered output clock CK_{REC} to generate data output D_{OUT}. The frequency of GVCO is controlled using the control voltage, V_C, obtained from a PLL with a replica of GVCO. Due to direct injection of data, GVCO-based architectures suffer from large output jitter. This is particularly problematic with high loss channels due to their large data dependent jitter (DDJ). Furthermore, design of limiting amplifier and gating circuit becomes difficult as data rates increase, resulting in excess power consumption.

Blind oversampling-based burst mode CDR, shown in Fig. 4.4, samples the input multiple times in a bit period using multi-phase samplers [73, 74, 75]. A digital phase and data picking logic block chooses the optimum data sample to provide recovered data, D_{OUT}. The multi-phase clock generator (MPG) provides the multiple clock phases for multi-phase samplers. Typically three to five samples are taken per bit period. As a result, oversampling-based CDR has much larger power consumption than conventional bang-bang phase detector

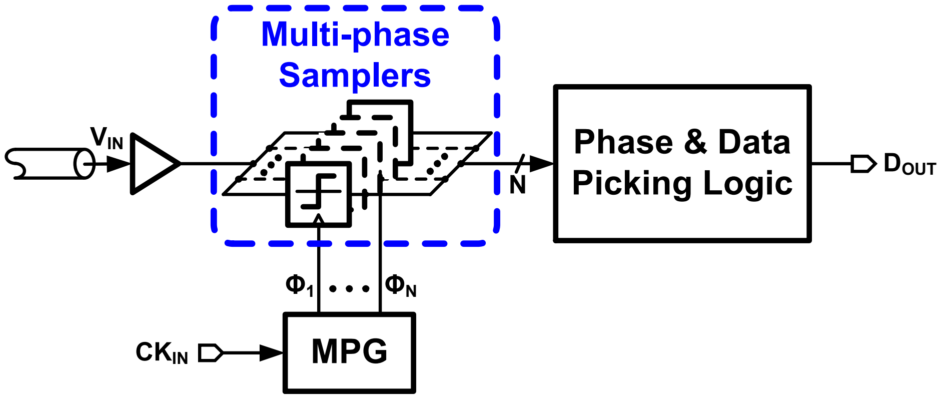


Figure 4.4: Simplified block diagram of oversampling-based CDR.

based CDR. This architecture is also not viable when significant equalization is needed for the receiver input, particularly in the form of discrete-time equalizers such as DFE and FFE.

An adaptive gain, bang-bang phase detector-based CDR is proposed in [76] achieving lock time of less than 20 ns. The effectiveness of this method is compromised in presence of large DDJ as the phase detector decisions become unreliable. Another technique proposed in [77] utilizes an extra phase rotator in conjunction with successive approximation logic to accurately set the initial phase of the CDR. Phase lock time of 18 ns is achieved using this method.

The burst-mode CDR techniques explored in the past implicitly assume that the DDJ on the input data is small, i.e. channels have low or moderate losses. These techniques have not been used in the context of high loss channels. Most of these techniques also require more than one sample per bit period for burst mode operation and implicitly assume 0.5 unit interval (UI) spacing between data transition edge and optimum data sampling point. In this work, we propose a highly digital burst-mode CDR technique that can be used with baud rate CDR architecture to achieve fast synchronization.

4.2 Principle of Proposed Burst-Mode Operation

Phase presetting, also called zero phase start, has been shown to reduce the synchronization time significantly [78, 79, 77]. Figure 4.5 shows a simplified block diagram of a phase presetting type burst mode CDR. To acquire initial phase lock, switch S_0 is closed and

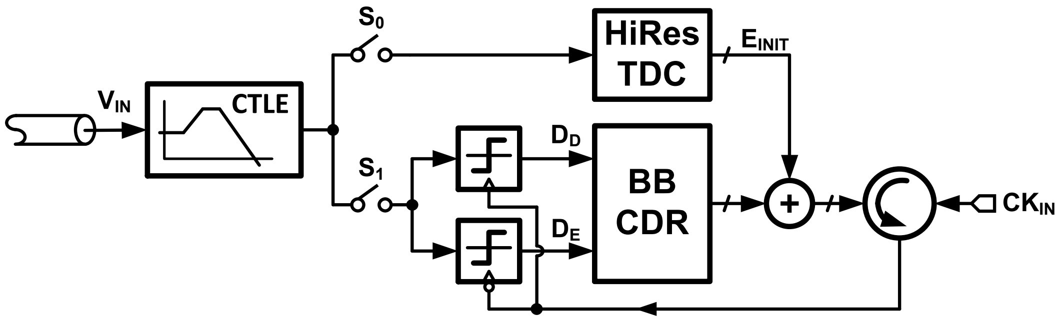


Figure 4.5: Simplified block diagram of phase presetting CDR.

switch S_1 is opened. A known preamble such as 0101 data pattern is used to find the digital representation, E_{INIT}, of the initial phase difference between data and clock using a high resolution time-to-digital converter (TDC). This initial phase difference is then added to the phase rotator and control is switched to the bang-bang CDR logic (BB CDR) for regular operation. A high resolution TDC consumes more power than bang-bang phase detector, therefore it is more efficient to use bang-bang phase detector during regular operation. On the other hand, the oversampling delta modulator-like nature of bang-bang operation requires more time to acquire phase lock. Phase presetting combines the strengths of both approaches to achieve low power during regular operation and fast synchronization in burst-mode. There are multiple ways of implementing the high resolution TDC efficiently. A two-step TDC is used as high resolution TDC in [79], whereas a successive approximation algorithm does high resolution time-to-digital conversion in [77].

To utilize phase presetting principle in the context of a baud rate receiver with discrete time equalization, we need to account for the sampled nature of FFE and DFE. While the continuous time linear equalizer can be considered time invariant, the discrete time DFE and FFE are time varying. As a result, incorrect sampling location for a given set of FFE and DFE coefficients may result in reduced sampling margins and, in the worst case, may close the input eye completely. To illustrate this issue, consider the channel pulse response shown in Fig. 4.6(a). The FFE/DFE coefficients are found from the ideal sampling location and the unequalized channel pulse response. In Fig. 4.6, t_0 denotes the ideal sampling location and T denotes the bit period. When the receiver samples input at the ideal sampling location, the resulting pulse response has zero ISI as shown in Fig. 4.6(b). On the other hand, any

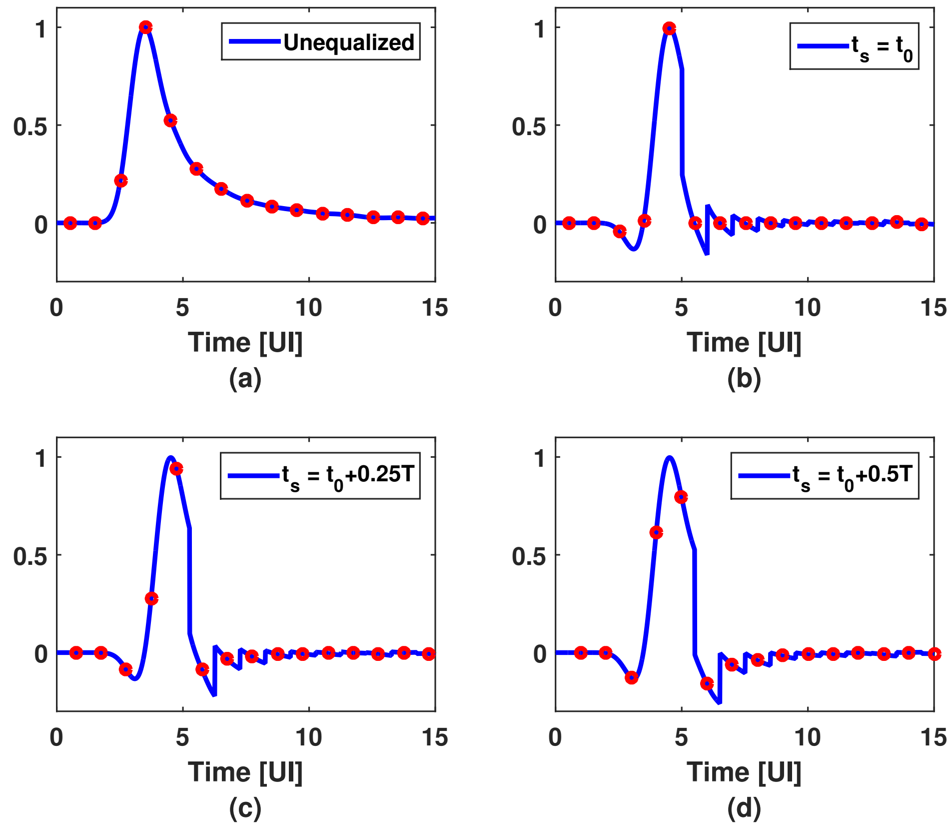


Figure 4.6: Channel pulse response in presence of baud rate discrete time FFE/DFE and timing offset.

sampling offset results in a pulse response with non-zero ISI (Fig. 4.6(c), and (d)). The sampler input eye diagrams for each of the above cases are shown in Fig. 4.7. As evident from Fig. 4.7, timing offsets in presence of FFE/DFE may result in worse sampling margins causing errors in data detection as well as phase detection. This indicates that the high resolution TDC transfer characteristic is a function of input data pattern as well as discrete time equalizer coefficients. We elaborate more on this in the following subsection.

4.2.1 Relation between Sampling Instant and Sampled Voltage

One of the methods of doing high resolution time-to-digital conversion is to first convert the time domain phase information into voltage and follow it with an ADC [80]. In the case of a receiver, the relationship between phase (or the sampling instant of input) and the sampled voltage value cannot be controlled. Instead we propose to estimate the channel

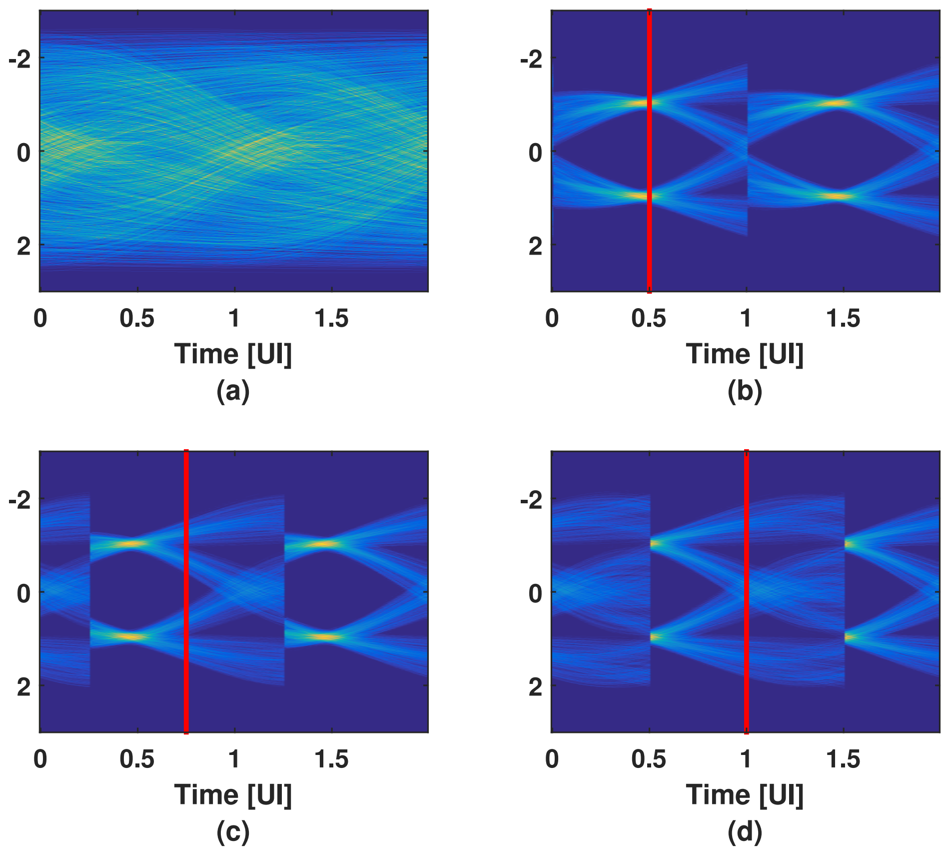


Figure 4.7: Sampler input eye diagrams for unequalized channel and in presence of FFE/DFE with various timing offsets. The vertical red line denotes the sampling location.

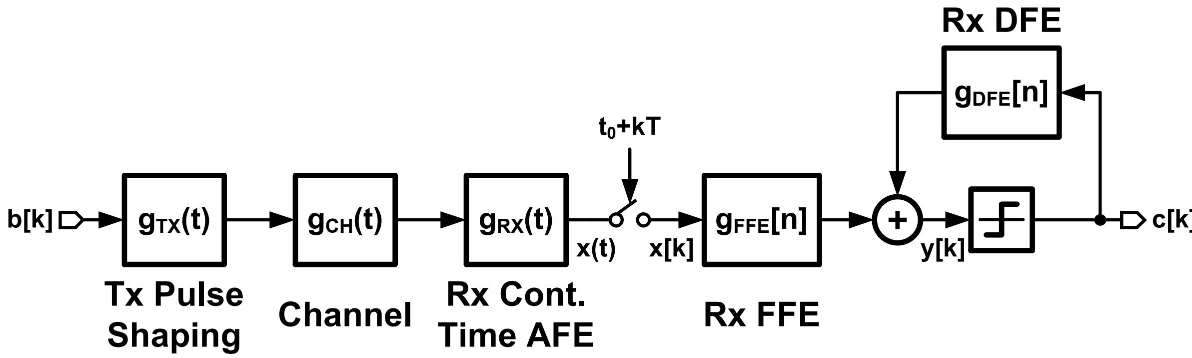


Figure 4.8: Simplified discrete time link model.

pulse response and use it to find the relationship between phase and sampled voltage value. Consider a simplified discrete time link model as shown in Fig. 4.8 [81]. We make two key assumptions in this model. The first is that all the blocks are linear, and the second is that the link is limited by ISI introduced by the band-limited channel and therefore we can ignore the effect of random noise. The input to the link is a binary sequence $b[k] \in \{\pm 1\}$. It is converted into a continuous time waveform using a transmitter pulse shaping filter whose pulse response is denoted as $g_{\mathrm{TX}}(t)$. The output of the transmitter is passed through a channel with impulse response $g_{\mathrm{CH}}(t)$. The impulse response of the continuous time analog front-end (AFE) of the receiver is denoted as $g_{\mathrm{RX}}(t)$. The output, $x(t)$, of the receiver AFE is sampled to generate a discrete time sequence $x[k]$. As before, the sampling time offset is denoted as t_0 and bit period is denoted as T. Note that $0 \leq t_0 \leq T$. The discrete time impulse response sequences for the receiver FFE and DFE are denoted as $g_{\mathrm{FFE}}[n]$ and $g_{\mathrm{DFE}}[n]$, respectively. The output of the DFE / FFE summer is denoted as y[k]. A 1-bit quantizer is used as a decision device to generate output sequence $c[k] \in \{\pm 1\}$ from $y[k]$. Let $h(t)$ denote the overall pulse response of the continuous time blocks. Then we have

$$h(t) = g_{\mathrm{TX}}(t) * g_{\mathrm{CH}}(t) * g_{\mathrm{RX}}(t) \tag{4.1}$$

The output of the receiver AFE is given by

$$x(t) = \sum_{n=-\infty}^{n=\infty} b[n]h(t - nT) \tag{4.2}$$

Due to sampling operation, we have

$$
\begin{aligned}
x[k] &= x(t_0 + kT) \\
x[k] &= \sum_{n=-\infty}^{n=\infty} b[n]h(t_0 + kT - nT) && (4.3) \\
y[k] &= (g_{\text{FFE}}[k] * x[k]) + (g_{\text{DFE}}[k] * c[k]) && (4.4) \\
c[k] &= sgn(y[k]) && (4.5)
\end{aligned}
$$

From Eq. 4.4 we see that $y[k]$ is a function of sampling time offset t_0. If $g_{\text{FFE}}[k]$, $g_{\text{DFE}}[n]$, $b[k]$, and $h(t)$ are known, we can find mapping from $y[k]$ to t_0.

$$
\begin{aligned}
y[k] &= f(k, t_0) && (4.6) \\
\Longrightarrow t_0 &= f^{-1}(k, y[k]) && (4.7)
\end{aligned}
$$

Note that the inverse mapping $f^{-1}(\cdot)$ is unique if $f(\cdot)$ is a one-to-one function. Most practical high speed wireline link channels satisfy this requirement. This means that by having a high resolution ADC to convert $y[k]$ into a digital code, we can achieve high resolution time-to-digital conversion. The coefficients $g_{\text{FFE}}[k]$ and $g_{\text{DFE}}[k]$ are known to the receiver. A preamble of known $b[k]$ can be agreed upon *a priori* for burst-mode operation. This leaves us with the problem of estimating the continuous time pulse response of the link, $h(t)$.

4.2.2 Link Pulse Response Estimation

To estimate the link pulse response, we observe that the FFE and DFE coefficients depend on the sampled link pulse response. Suppose the sampled link pulse response is given by

$$
\begin{aligned}
h_{t_0}[k] &= h(t_0 + kT) && (4.8) \\
H_{t_0}(z) &= \mathcal{Z}\{h_{t_0}[k]\} && (4.9)
\end{aligned}
$$

where $\mathcal{Z}\{\cdot\}$ denotes the z-transform operator. We can further simplify the discrete time link model as shown in Fig. 4.9. We have assumed that there are no errors in the link. The

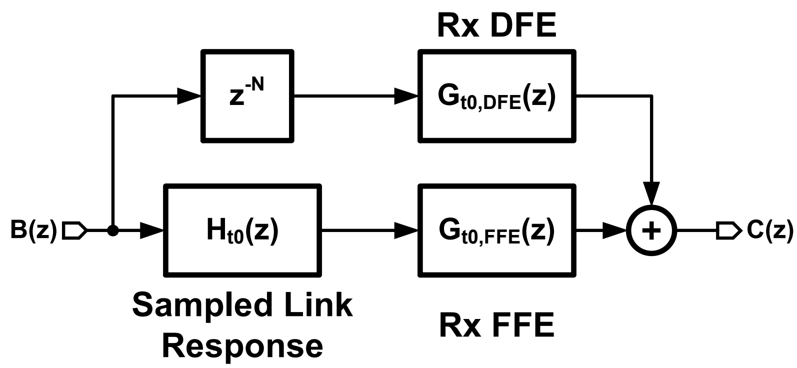


Figure 4.9: Simplified z-domain link model.

overall link transfer function is given by

$$\frac{C(z)}{B(z)} = (H_{t_0}(z)G_{t_0,\mathrm{FFE}}(z)) + \left(z^{-N}G_{t_0,\mathrm{DFE}}(z)\right) \tag{4.10}$$

where $G_{t_0,\mathrm{FFE}}(z)$, $G_{t_0,\mathrm{DFE}}(z)$, and N denote FFE transfer function, DFE transfer function, and channel delay, respectively. If $G_{t_0,\mathrm{FFE}}(z)$, and $G_{t_0,\mathrm{DFE}}(z)$ are chosen so as to achieve zero forcing equalization (ZFE), we would have

$$\begin{aligned} \frac{C(z)}{B(z)} &= z^{-N} \\ \Longrightarrow H_{t_0}(z) &= z^{-N} \cdot \frac{1 - G_{t_0,\mathrm{DFE}}(z)}{G_{t_0,\mathrm{FFE}}(z)} \end{aligned} \tag{4.11}$$

As the channel delay, N, does not play any important role in receiver synchronization, we will ignore it for the subsequent analysis. Therefore, the sampled link pulse response can be found as

$$h_{t_0}[k] = \mathcal{Z}^{-1}\{H_{t_0}(z)\} = \mathcal{Z}^{-1}\left\{\frac{1 - G_{t_0,\mathrm{DFE}}(z)}{G_{t_0,\mathrm{FFE}}(z)}\right\} \tag{4.12}$$

where $\mathcal{Z}^{-1}\{\cdot\}$ denotes the inverse z-transform operator.

In practice, the equalizer coefficients are found using an adaptive algorithm such as least mean squares (LMS). The coefficients found using LMS algorithm approximately satisfy the minimum mean square error (MMSE) criterion. Note that coefficients that satisfy MMSE criterion in case of an ISI-limited link, also satisfy ZFE criterion. Therefore, the most straightforward method to estimate $h(t)$ is to find the receiver equalizer coefficients at every sampling offset $0 \leq t_0 \leq T$ using LMS algorithm. This is impractical and cannot be done without disrupting regular receiver operation. Instead, we turn to the band-limited nature of the channel and propose the use of interpolation for estimation of link pulse response $h(t)$. This is similar to the reconstruction filter mechanism typically used at DAC output to convert a discrete time sequence to a smooth continuous time waveform. If $s(t)$ denotes the impulse response of a first order sample and hold function, and $g_{\text{INT}}(t)$ denotes the impulse response of a interpolation filter, then

$$h_{\text{est}}(t) = \left(\sum_{n=-\infty}^{n=\infty} h_{t_0}[n] s(t - t_0 - nT) \right) * g_{\text{INT}}(t) \tag{4.13}$$

By using interpolation, estimation of the link pulse response can be done in the background. To maintain the accuracy of interpolation, it must be ensured that the bandwidth of the interpolation filter is larger than the link bandwidth. Alternatively, multiple samples per bit period may be necessary if the link bandwidth is much larger than the interpolation filter bandwidth. We illustrate the above procedure using an example in the next subsection.

4.2.3 Illustrative Example

For an illustrative example, we use a 29.8″ long Megtron 6 PCB channel whose S-parameters are available at [82]. We assume the data rate to be 25.6 Gb/s. The transmitter pulse shape is assumed to be a rectangle. All the simulations for this example are carried out using MATLAB. The transfer function of the channel is shown in Fig. 4.10. The channel loss at the Nyquist frequency of 12.8 GHz is 18.6 dB. On the receiver side we assume that a two-tap FFE and a 10-tap DFE are present. Figure 4.11 shows the sampled link pulse response, $h_{t_0}[k]$, and the estimated sampled link pulse response, $h_{t_0,\text{est}}[k]$. The estimated pulse response is calculated using the FFE/DFE coefficients found using LMS algorithm and Eq. 4.12. The estimated pulse responses using interpolation methods such as linear interpolation and shape-preserving piecewise cubic (pchip) interpolation are shown along with actual link pulse response in Fig 4.12. A reasonable matching is found between estimated pulse responses and actual pulse response. The link pulse response can be used to calculate the expected

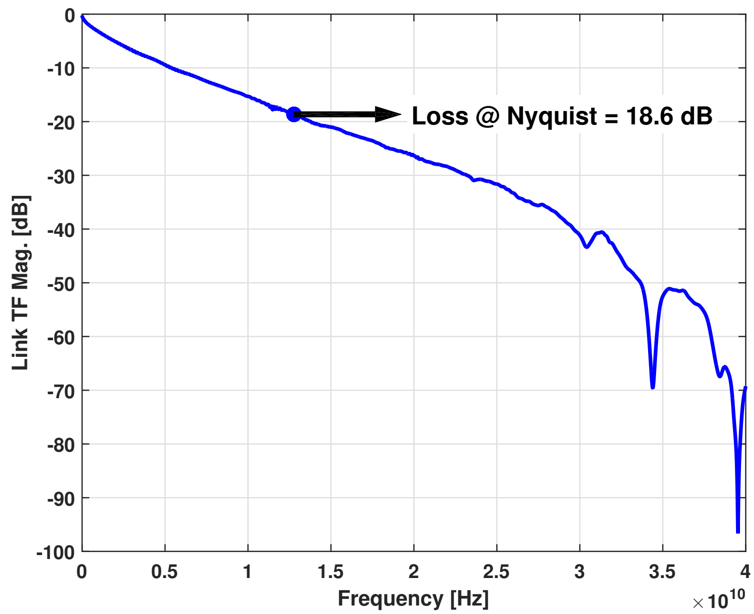


Figure 4.10: An example channel transfer function [82].

waveform for a -1 to +1 rising edge of the waveform for a {-1, -1, +1, +1} repetitive symbol pattern. Figure 4.13 indicates good agreement between actual link edge response and estimated link edge responses. This edge response can be used as a mapping function from voltage to phase, opening up a possibility of high resolution time-to-digital conversion in presence of large channel losses.

4.2.4 Burst-Mode Operation

To implement high resolution phase detection necessary for fast synchronization, we propose the following sequence:

1. Transmitter starts transmission with a known preamble symbol sequence such as {-1, -1, +1, +1}.
2. The receiver detects -1 to +1 transition edge and digitizes the sampled voltage value.
3. A lookup table containing estimated phase to voltage mapping is used to find out the phase offset based on the digitized sampled voltage value.

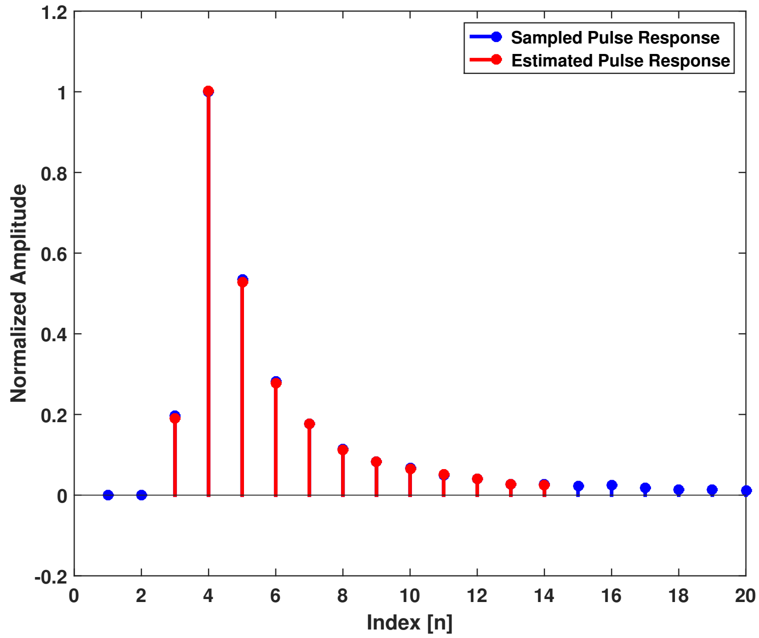


Figure 4.11: Sampled link pulse response and recovered pulse response using FFE/DFE coefficients.

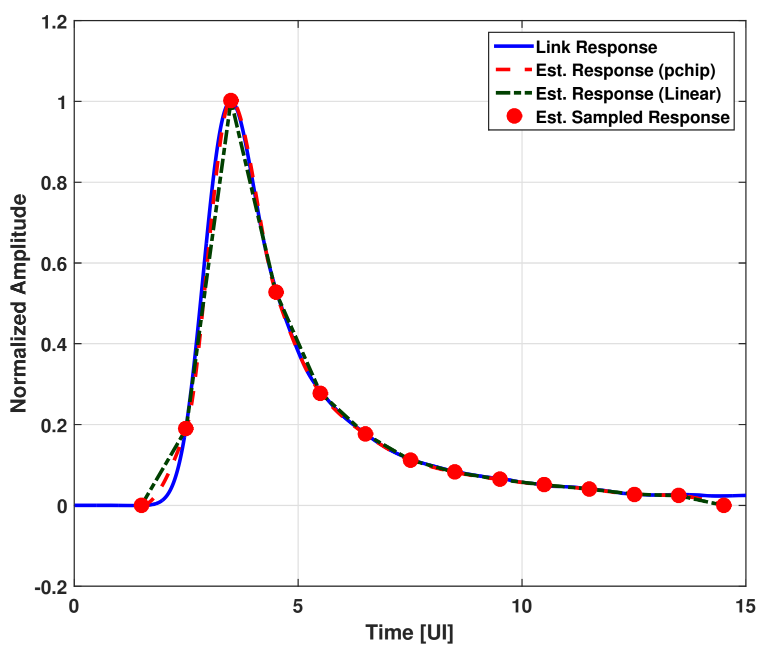


Figure 4.12: Link pulse response and estimated link pulse responses using interpolation.

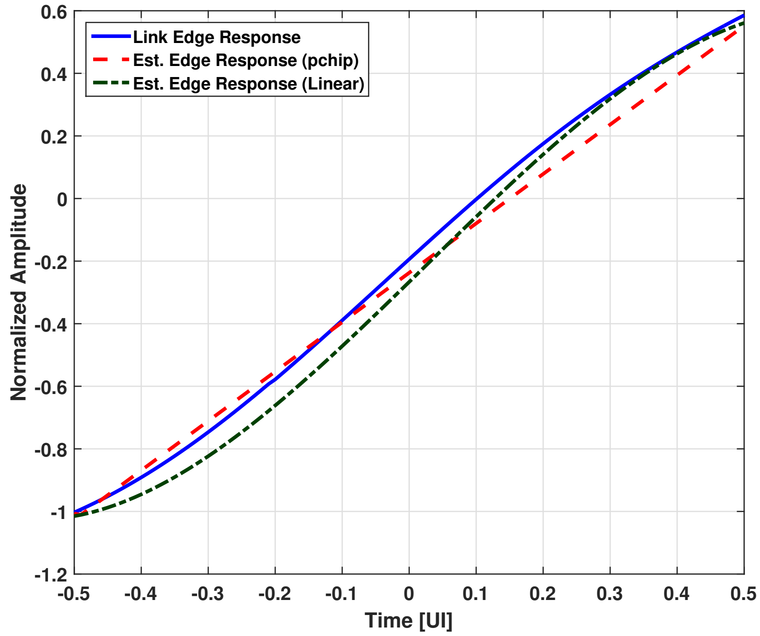


Figure 4.13: Link edge response and estimated link edge responses using interpolation for a -1 to +1 transition edge of a {-1, -1, +1, +1} symbol pattern.

4. A phase rotator is used to apply the appropriate amount of phase offset so that the receiver starts sampling the input at the optimum sampling location.

It should be noted that the analog-to-digital conversion of the input voltage need not happen at data rate. This relaxes the speed as well as power requirement for the analog-to-digital conversion circuitry, albeit at the cost of increased synchronization latency. In next section, we describe techniques to implement the principle of burst-mode operation discussed in this section.

4.3 Receiver Implementation

A burst mode receiver is implemented in 65nm CMOS technology to demonstrate the principle of operation described previously. Figure 4.14 depicts the top level architecture of the receiver. The continuous time AFE consists of a CTLE followed by a variable gain amplifier (VGA). A quarter rate architecture is chosen for the discrete equalizer to reduce the clocking frequency. Each equalizer slice consists of a two-tap FFE and two-tap DFE. The interconnections between the slices are not shown for simplicity. Baud rate CDR architec-

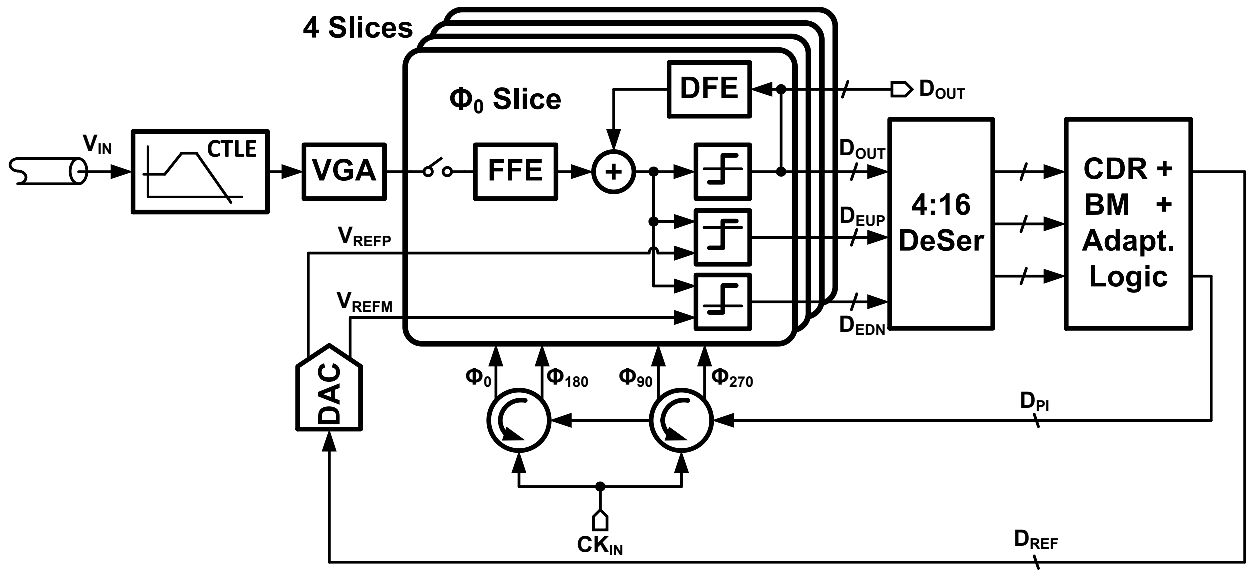


Figure 4.14: Simplified receiver block diagram.

ture also reduces the number of clock phases required for implementation. The outputs of all the slices are deserialized using a 4:16 deserializer. The outputs of the deserializer go to a digital logic block that combines Mueller-Muller CDR logic, burst-mode operation logic (BM), and reference as well as equalizer coefficient adaptation logic. This block controls the phase rotator input D_{PI}, reference DAC input D_{REF} as well as equalizer coefficients. The quarter rate architecture makes use of two phase rotators that provide four clock phases, Φ_0, Φ_{90}, Φ_{180}, and Φ_{270}. We will describe the key implementation details for the receiver next.

4.3.1 Front-End Amplifiers

The receiver utilizes a conventional CTLE architecture with source degenerated RC network as shown in Fig. 4.15. The resistance and capacitance of the RC network can be varied to achieve programmable de-emphasis. The output of the CTLE drives a variable gain amplifier (VGA). VGA gain can be varied by changing its source degeneration resistance. The VGA drives the switched load presented by track and hold (T/H) circuits in equalizer slices. To be able to drive the large parasitic capacitance, inductive peaking is used in the VGA. The simulated AC response of the CTLE followed by VGA is shown in Fig. 4.16. Under typical conditions, the front-end amplifiers can provide a maximum peaking of around 15 dB at

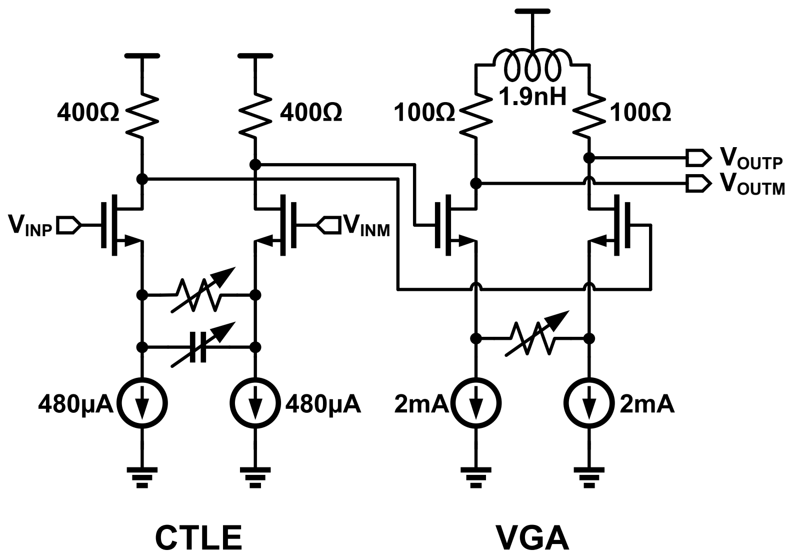


Figure 4.15: Simplified front-end amplifier circuit diagram.

frequency of 4 GHz. The peaking frequency of 4 GHz for front-end amplifiers corresponds to a quarter of baud rate at 16Gb/s and is suggested to be optimal when CTLE is used in conjunction with DFE [83]. Simulations indicate that the front-end amplifiers consume around 5 mW power while driving a load capacitance of 100 fF. The CTLE consumes around 1 mW while the rest is consumed by the VGA.

4.3.2 Track and Hold Circuit

A T/H circuit is necessary to hold the input voltage value for multiple bit periods for FFE operation. In the receiver, the PMOS-based T/H circuit shown in Fig. 4.17 is adopted from [52]. A reset switch is added to the T/H circuit in order to reduce ISI at the output. Figure 4.18 shows the logic diagram of for the clocking circuit that generates 25% duty cycle clocks for the T/H circuit. An additional input, VDD_{TUNE}, can be used to vary the duty cycle of the T/H clock. Based on simulation, T/H clocking circuits are estimated to consume around 6 mW power under typical conditions while operating at 16 Gb/s data rate.

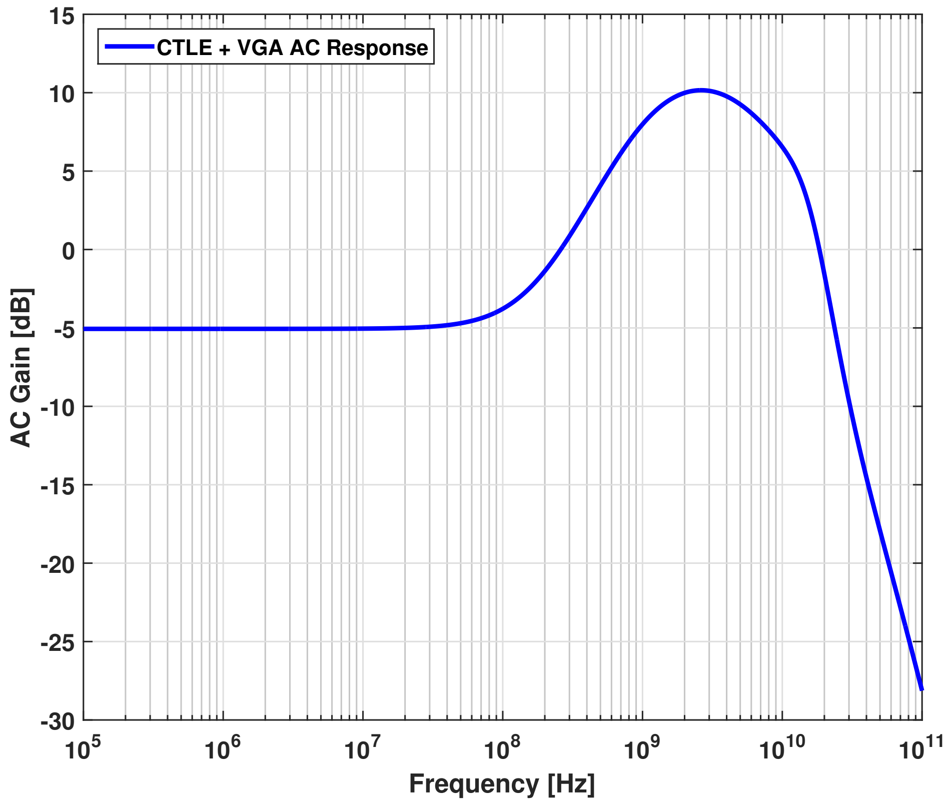


Figure 4.16: Simulated AC response of CTLE with VGA for maximum peaking.

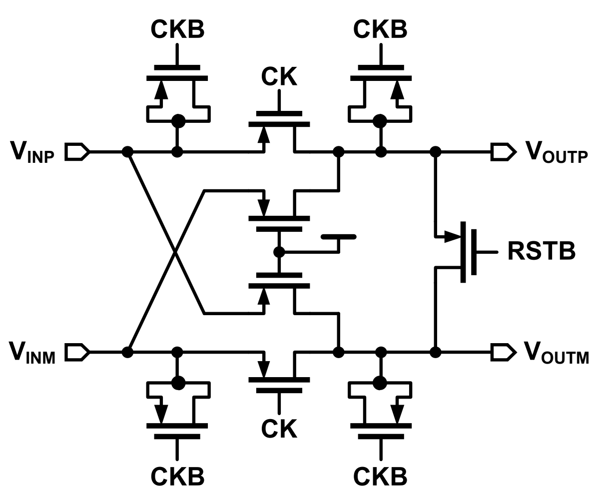


Figure 4.17: Simplified track and hold (T/H) circuit diagram.

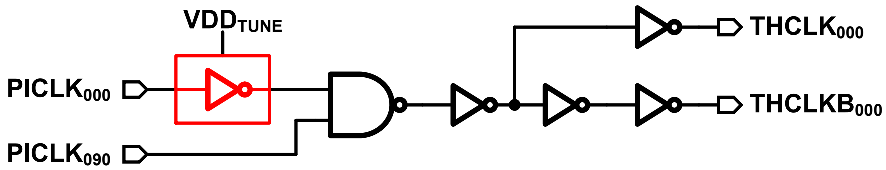


Figure 4.18: Simplified logic diagram for generating clock signal for T/H circuit with 25% duty cycle.

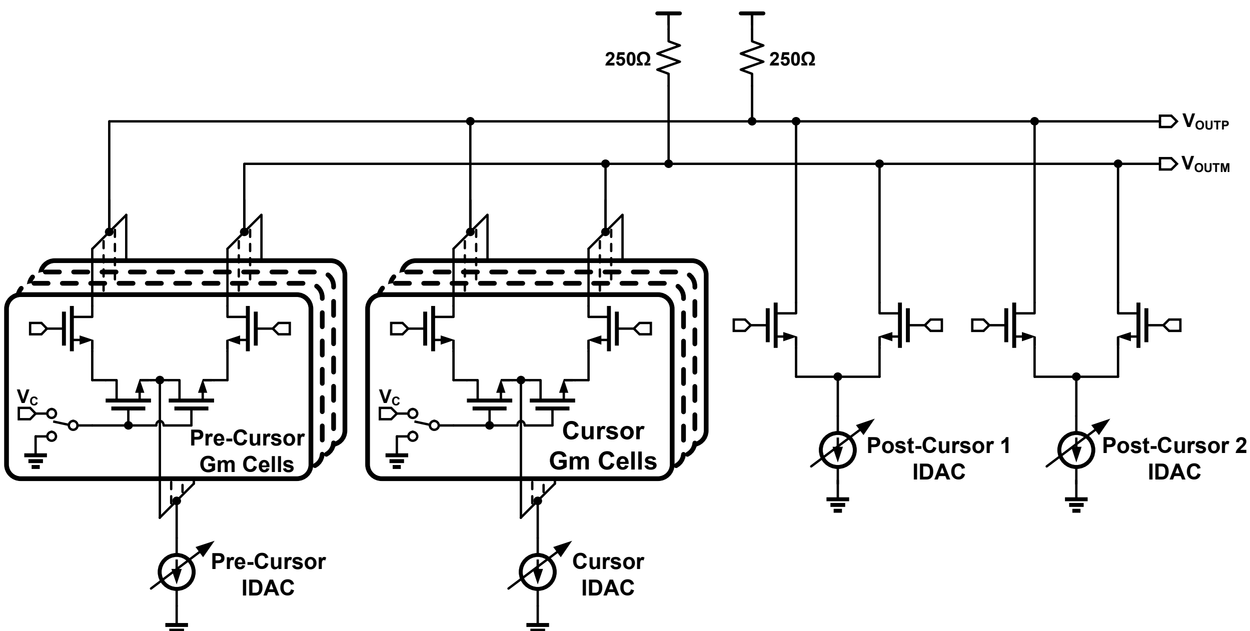


Figure 4.19: Simplified FFE/DFE summer circuit diagram.

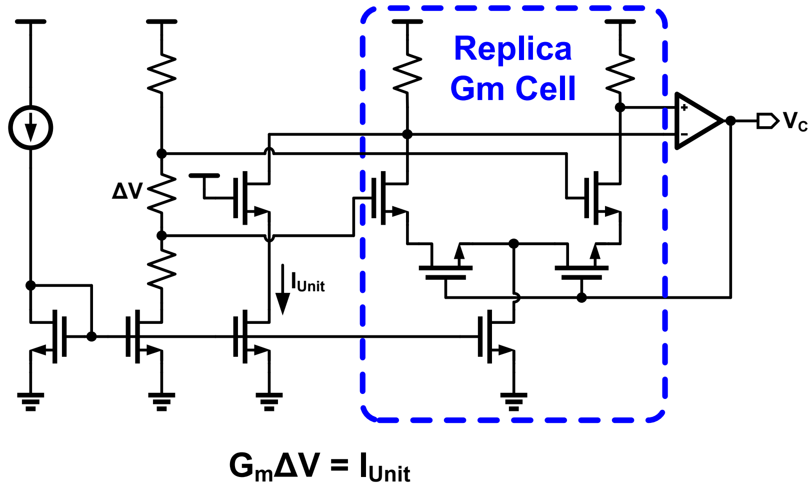


Figure 4.20: G_m control circuit for maintaining a known relationship between FFE and DFE coefficients.

4.3.3 FFE / DFE Summer

As mentioned before, the FFE / DFE implementation consists of two FFE taps and two DFE taps. Figure 4.19 shows the simplified circuit diagram of the summer. For FFE taps, programmable G_m cells are implemented using source degenerated differential pairs and tail current DACs (IDACs). Source degeneration is provided by MOS transistors operating in triode region. Compared to [84], the proposed approach for implementing FFE taps results in lower parasitics in the high speed path. The G_m of unit cell is controlled by control voltage, V_C. For pulse response estimation, a known relationship must exist between DFE coefficients and FFE coefficients. A G_m control circuit based on [85] that generates V_C to maintain a fixed relationship between FFE and DFE coefficients is shown in Fig. 4.20.

The equalizer coefficients are adapted so as to keep the total IDAC current at 1.6 mA. Table 4.1 tabulates the coefficient range and resolution for each of the cursors. Note that the unit current for FFE coefficients is higher than that for DFE coefficients as complete current steering is not possible with FFE unit G_m cells due to linearity constraints.

To relax the DFE feedback timing constraint, soft decision architecture is adopted for the FFE / DFE summer [86]. Figure 4.21 shows the simplified block diagram of the FFE / DFE. Clock signals with 25% duty cycle ($CK_{TH,000}$, $CK_{TH,090}$, $CK_{TH,180}$, and $CK_{TH,270}$) are used to sample the input voltage, V_{IN}. The same clock signals also act as active low reset signals

Table 4.1: FFE / DFE coefficient range and resolution.

	Min. Current	Max. Current	Current Res.	Bits
Pre-Cursor	0	800 μA	50 μA	4
Main Cursor	800 μA	1600 μA	50 μA	4
Post Cursor 1	0	320 μA	10 μA	5
Post Cursor 2	0	160 μA	10 μA	4

for T/H circuit in the neighboring slice. Separate quarter rate clock signals with 50% duty cycle ($\mathrm{CK_{LAT,000}}$, $\mathrm{CK_{LAT,090}}$, $\mathrm{CK_{LAT,180}}$, and $\mathrm{CK_{LAT,270}}$) are used for clocking the latches at the output of the summer. The outputs of the latches ($\mathrm{D_{OUT,000}}$, $\mathrm{D_{OUT,090}}$, $\mathrm{D_{OUT,180}}$, and $\mathrm{D_{OUT,270}}$) drive the DFE inputs as well as another set of latches that are not shown for simplicity.

4.3.4 Digital Logic

The digital logic block performs Mueller-Muller clock recovery using the deserialized data. A second order digital CDR loop is implemented that can track booth phase and frequency variations using only phase rotators [87]. Sign-sign LMS algorithm [88, 89] is used for adapting the FFE and DFE coefficients as well as reference voltage levels of error samplers.

4.3.5 Edge Response Estimation

Recursion based formulation is used to implement edge response estimation described in Section 4.2.2. The first step is to estimate the link pulse response from equalizer coefficients. For the present implementation we have two FFE taps and two DFE taps. Let

$$\begin{aligned} \mathbf{b[k]} &= \begin{bmatrix} b[k+1] & b[k] & b[k-1] & b[k-1] \end{bmatrix}^T \\ \mathbf{h_{t_0}} &= \begin{bmatrix} h_{t_0}[-1] & h_{t_0}[0] & h_{t_0}[1] & h_{t_0}[2] \end{bmatrix}^T \end{aligned}$$

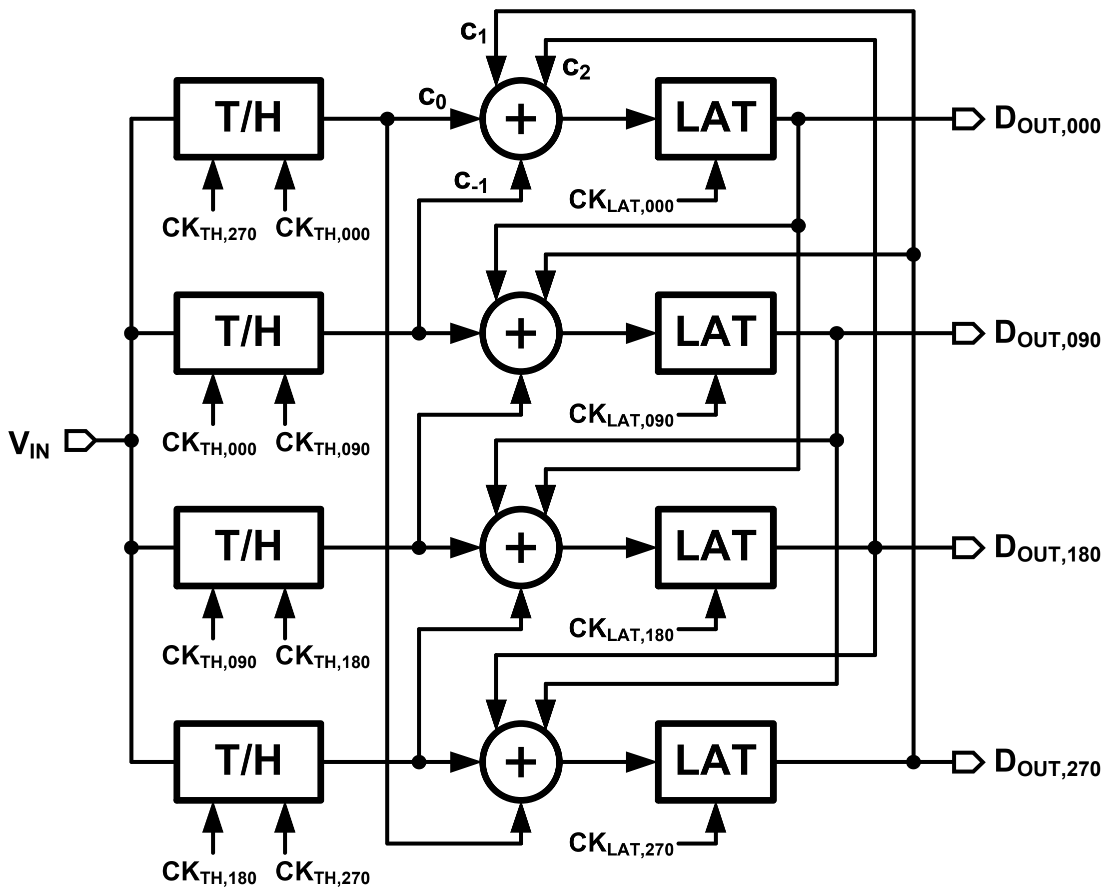


Figure 4.21: Simplified FFE / DFE block diagram.

where $b[k]$ are the input symbols and $h_{t_0}[k]$ are sampled link pulse response values at sampling offset t_0. The input to the equalizer is given by

$$x[k] = \mathbf{h_{t_0}}^T \mathbf{b[k]}$$

Let A be the amplitude estimated from the reference adaptation loop. Let $\mathbf{c}_{\text{FFE}} = [c_{-1}\ c_0]^T$ be the estimated FFE coefficients and $\mathbf{c}_{\text{DFE}} = [c_1\ c_2]^T$ be the estimated DFE coefficients. Assuming perfect zero-forcing equalization,

$$Ab[k] = \mathbf{c}_{\text{FFE}}{}^T \begin{bmatrix} x[k+1] & x[k] \end{bmatrix}^T + \mathbf{c}_{\text{DFE}}{}^T \begin{bmatrix} b[k-1] & b[k-2] \end{bmatrix}^T$$

The simultaneous equations above can be solved to arrive at the following expressions:

$$\begin{aligned}
h_{t_0}[2] &= -\frac{c_2}{c_0} \\
h_{t_0}[1] &= -\frac{c_1}{c_0} - \frac{c_{-1}h_{t_0}[2]}{c_0} \\
h_{t_0}[0] &= \frac{A}{c_0} - \frac{c_{-1}h_{t_0}[1]}{c_0} \\
h_{t_0}[-1] &= -\frac{c_{-1}h_{t_0}[0]}{c_0}
\end{aligned}$$

Linear interpolation of the sampled link pulse response estimates is used to compute the response of the link to -1 to +1 transition for a preamble sequence of {-1,-1,+1,+1}. Since the resolution of the phase rotator in the receiver is 5-bits, let $v[n]$ denote the estimated voltage at phase offset $\frac{nT}{32}$, as shown in Fig. 4.22. For a symbol sequence

$$\{b[-2], b[-1], b[0], b[1], b[2], b[3]\} = \{-1, +1, +1, -1, -1, +1\}$$

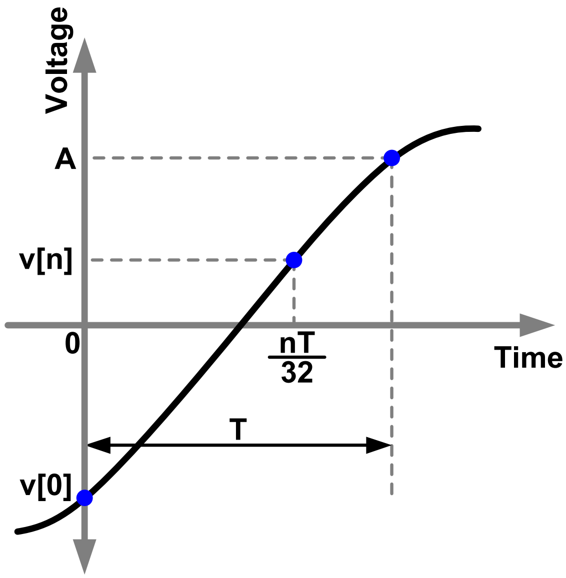


Figure 4.22: Illustration of -1 to +1 transition edge.

we can write

$$\begin{aligned} v[n] \quad = \quad & c_{-1}\left(h_{t_0}[-1] + h_{t_0}[0] - h_{t_0}[1] - h_{t_0}[2]\right) + \frac{2n}{32}c_{-1}\left(-h_{t_0}[-1] + h_{t_0}[1]\right) + \\ & c_0\left(h_{t_0}[-1] - h_{t_0}[0] - h_{t_0}[1] + h_{t_0}[2]\right) + \frac{2n}{32}c_0\left(h_{t_0}[0] - h_{t_0}[2]\right) + \\ & -c_1 - c_2 \end{aligned} \tag{4.14}$$

As the DAC resolution is 6-bits, a 6-b$\times$5-b look-up table (LUT) is used to store $v[n]$ values computed using Eq. 4.14. This LUT calculation can be done at a much lower rate based on the rate of variation of equalizer coefficients. A finite state machine (FSM) based implementation is used to share computational resources among various arithmetic operations. The estimated power and area of the edge response estimation block are given Table 4.2.

4.3.6 Burst-Mode Operation

During burst-mode operation, the transmitter sends a preamble sequence {-1, -1, +1, +1} while the receiver executes the following steps:

1. Select the slice with -1 to +1 transition in the quarter rate equalizer.

Table 4.2: Post-synthesis area and power consumption of edge response estimation block.

Technology	65 nm CMOS
Clock Frequency	7.8125 MHz
No. of Comb. Cells	3331
No. of Seq. Cells	592
Total Cell Area	16118.8 μm^2
Cell Internal Power	0.034 mW
Net Switching Power	0.027 mW
Leakage Power	0.076 mW
Total Power	0.137 mW

2. Do successive approximation-based analog-to-digital (A/D) conversion with slice that has -1 to +1 transition.

3. Find the phase offset using result of A/D conversion as well as LUT and apply the phase offset code to PI.

The flow chart for the burst-mode logic state machine is shown in Fig. 4.23.

To select the slice with -1 to +1 transition, we freeze the DFE feedback to {-1, -1}. By doing so, it is ensured that only one of the slices which receives the -1 to +1 transition has low error sampler output while its next neighboring slice has high error sampler output as shown in Fig. 4.24. The digital burst-mode logic state machine detects this transition in error sampler output. The samplers as well as reference DAC used to generate V_{REF} are the same as the ones used for Mueller-Muller CDR logic.

A successive register approximation register (SAR) based method is used for A/D conversion. The error sampler for CDR and reference DAC are reused for successive approximation-based A/D conversion during burst-mode operation as shown in Fig. 4.25. This obviates the need for any additional analog circuitry during burst-mode operation. The 4:16 deserializer is bypassed during A/D conversion to reduce the feedback latency of A/D conversion. The estimated power and area of the burst-mode digital logic block are given in Table 4.3.

The receiver chip is fabricated in 65 nm CMOS technology and occupies an area of 1.5 mm$\times$ 1.2 mm. The chip layout is shown in Fig. 4.26.

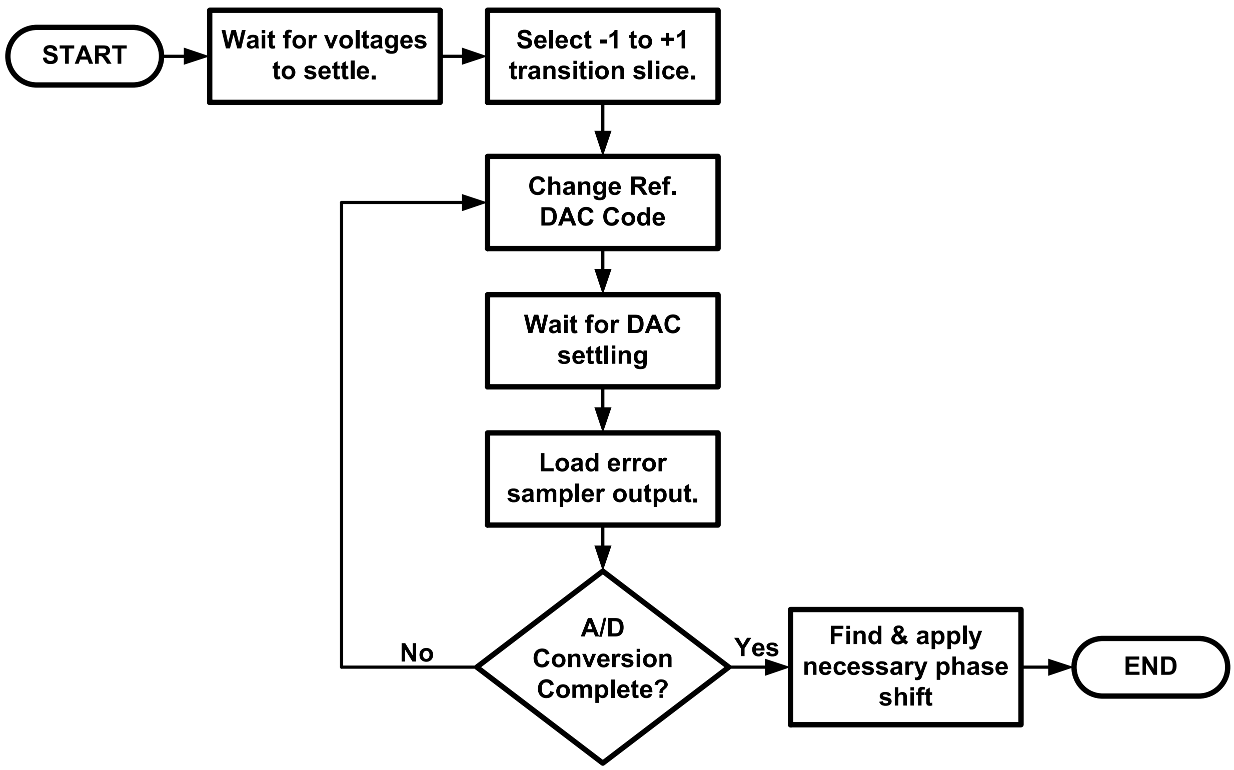


Figure 4.23: Flow chart for burst-mode logic state machine.

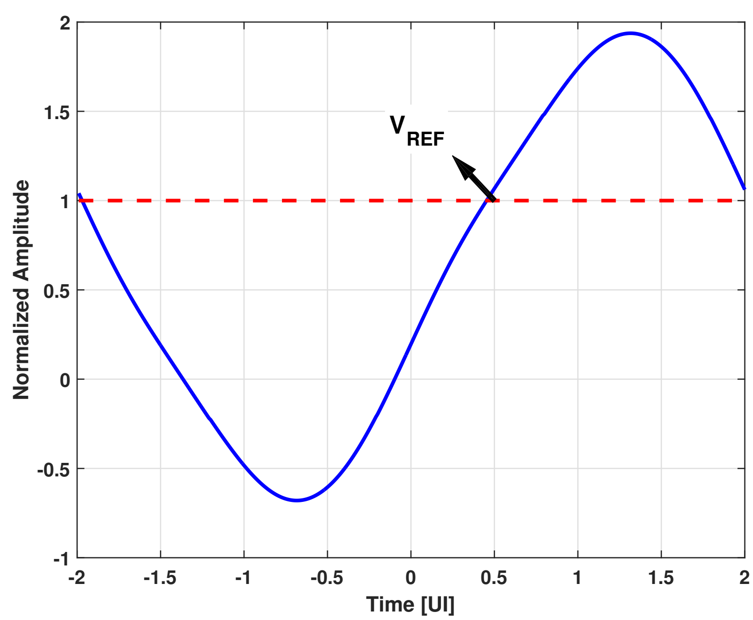


Figure 4.24: An example waveform for {-1, -1, +1, +1} pattern when DFE feedback is frozen to {-1, -1}.

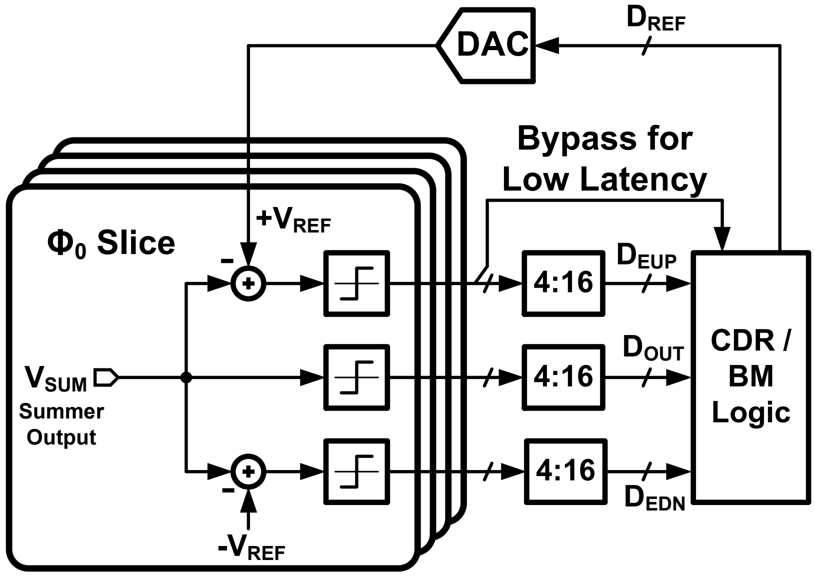


Figure 4.25: A/D conversion using error sampler and reference DAC.

Table 4.3: Post-synthesis area and power consumption of edge response estimation block.

Technology	65 nm CMOS
Clock Frequency	1 GHz
No. of Comb. Cells	1315
No. of Seq. Cells	155
Total Cell Area	9856 μm^2
Cell Internal Power	1.285 mW
Net Switching Power	0.567 mW
Leakage Power	0.290 mW
Total Power	2.142 mW

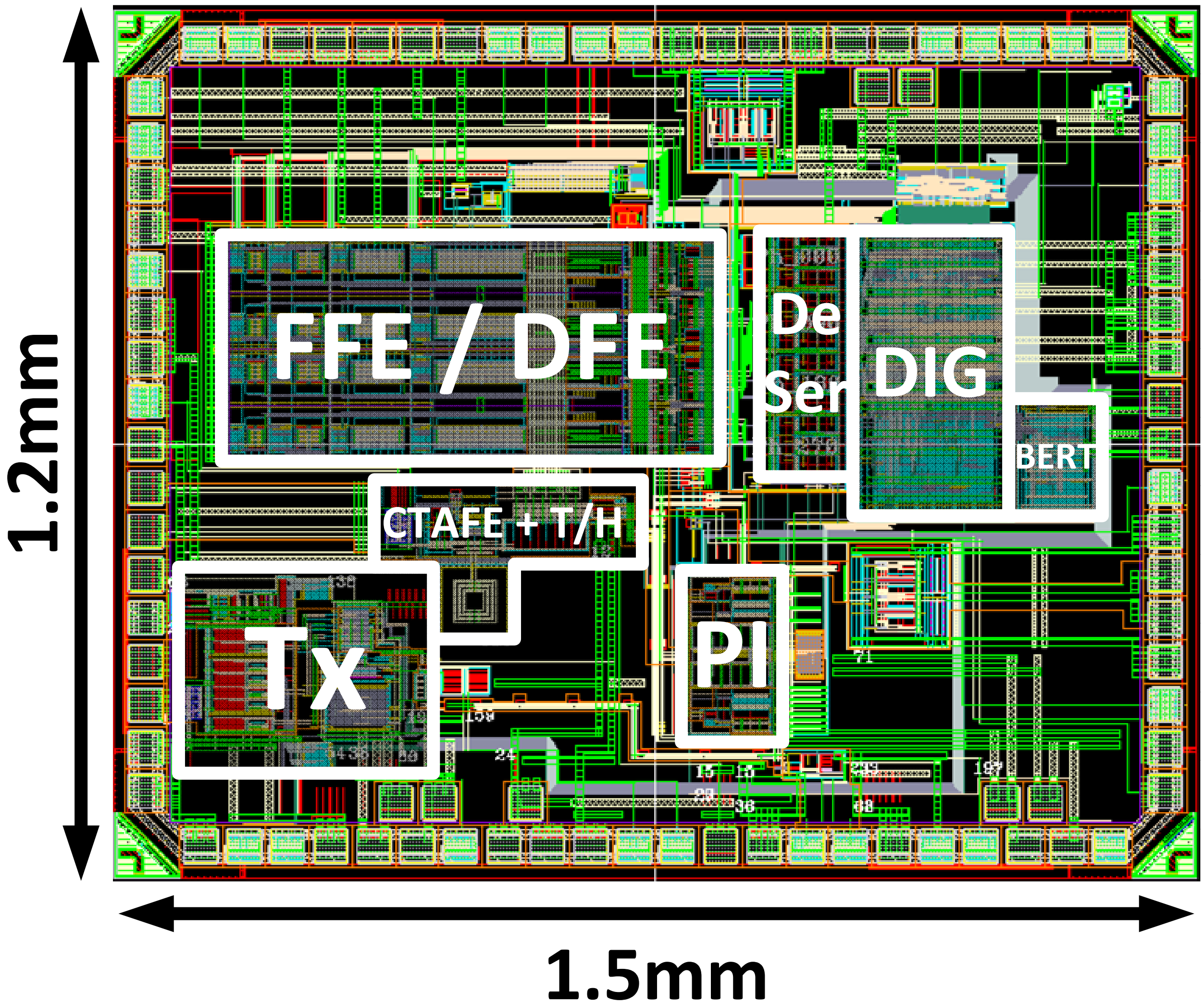


Figure 4.26: Receiver chip layout.

4.4 Conclusion

In this chapter we proposed a phase presetting-based burst mode receiver technique that utilizes link pulse response estimation to reduce the synchronization time. We also described implementation of a baud-rate sampling burst-mode receiver to demonstrate the efficacy of the proposed technique. The receiver, designed to operate at the data rate of 16 Gb/s, is implemented in 65 nm CMOS process technology. The highly digital nature and low analog hardware overhead of the proposed technique make it suitable for ADC-based links as well as links that utilize more complex modulation schemes such as 4-level pulse amplitude modulation (PAM4).

CHAPTER 5

A LOW COMPLEXITY LINK PULSE RESPONSE ESTIMATION TECHNIQUE

With the phenomenal advances in semiconductor processing, computer architecture and communication technology, we find ourselves in an era characterized by Big Data and ubiquitous computing. Data bandwidth requirements of computing platforms are rapidly growing, while miniaturization and cost factors are enforcing limits on power consumption. Increasingly the performance of the computing platforms is being limited by their data transfer bandwidth capabilities. This has given rise to the design of low power, high data rate wireline I/O interfaces [1]. These interfaces are characterized by the use of moderate loss channels, as well as sparing use of equalization to save power. In this chapter, we describe a low complexity, low overhead pulse response estimation technique that can be used for the characterization of low-power high-speed I/O interfaces where discrete time equalizers such as decision feedback equalizer (DFE) are not used.

This chapter is organized as follows. We provide an overview of channel pulse response in Section 5.1. Section 5.2 briefly summarizes the important adaptation and link characterization techniques that are used in high speed wireline I/O interfaces. It also describes the low complexity pulse response characterization technique. Details of implementation and simulation results are shown in Section 5.3. We conclude this chapter in Section 5.4.

5.1 Background

5.1.1 Channel Losses and Pulse Response

At high data rates, wireline channels such as printed circuit board (PCB) traces and copper cables exhibit losses and dispersion due to phenomena such as skin effect and dielectric

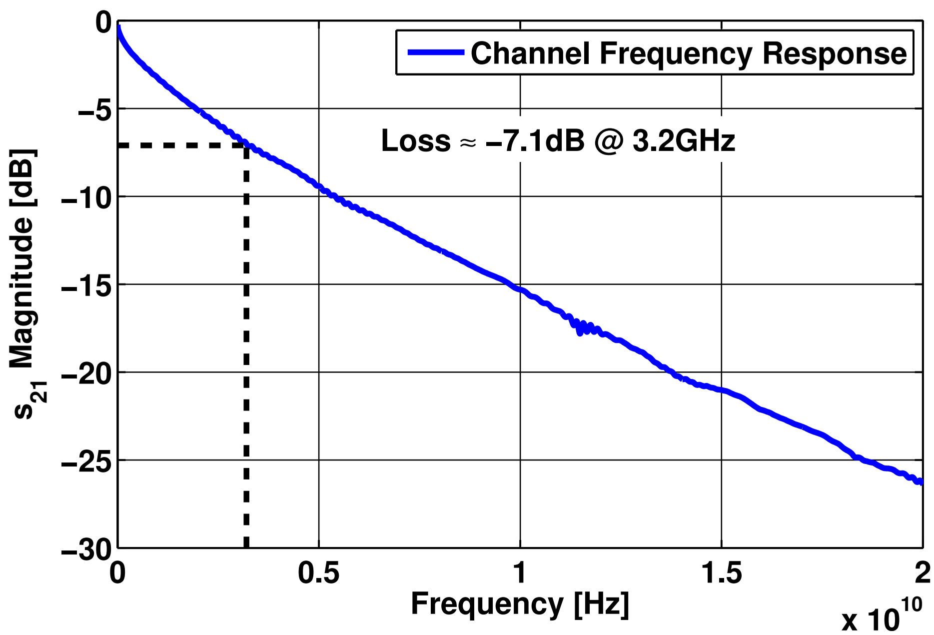


Figure 5.1: Frequency response of a 29.8″ long Megtron-6 channel [82].

loss. These result in inter-symbol interference (ISI) that hinders error-free detection of data. Channel loss is typically reported at Nyquist frequency. For example, loss of a 29.8″ long channel using Megtron-6 PCB material [82] is about 7.1 dB at 3.2 GHz as shown in Fig. 5.1. Corresponding pulse response and eye diagram at the data rate of 6.4 Gb/s are shown in Fig. 5.2 and Fig. 5.3 respectively. The pulse response of a channel is an important tool for link diagnostics as well as statistical performance evaluation of the link [90]. While the pulse response of a channel can be derived from its frequency response, it is not always feasible to measure channel frequency response. Further, many non-linear effects due to output driver circuit may not be captured by frequency response. On the other hand channel pulse response is a time domain measure and can capture non-linear effects. With these considerations in mind, it is desirable to estimate the channel pulse response.

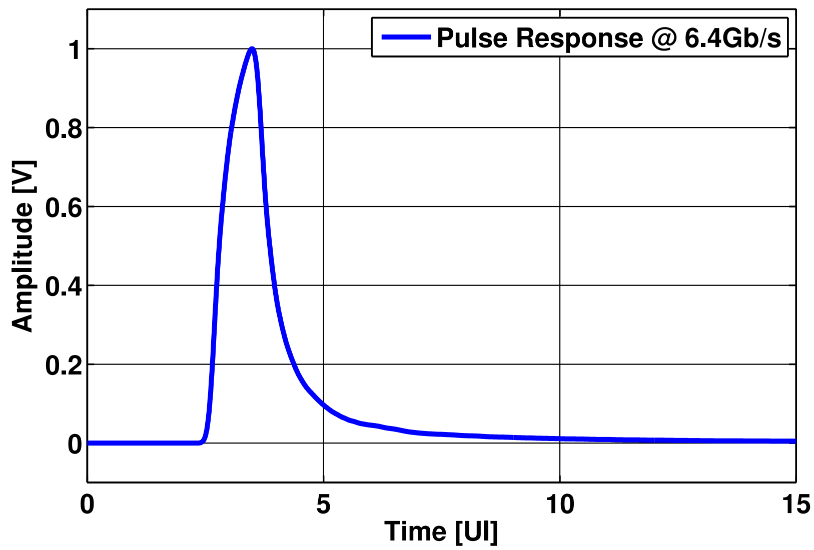


Figure 5.2: Pulse response of a 29.8″ long Megtron-6 channel [82] at 6.4 Gb/s data rate.

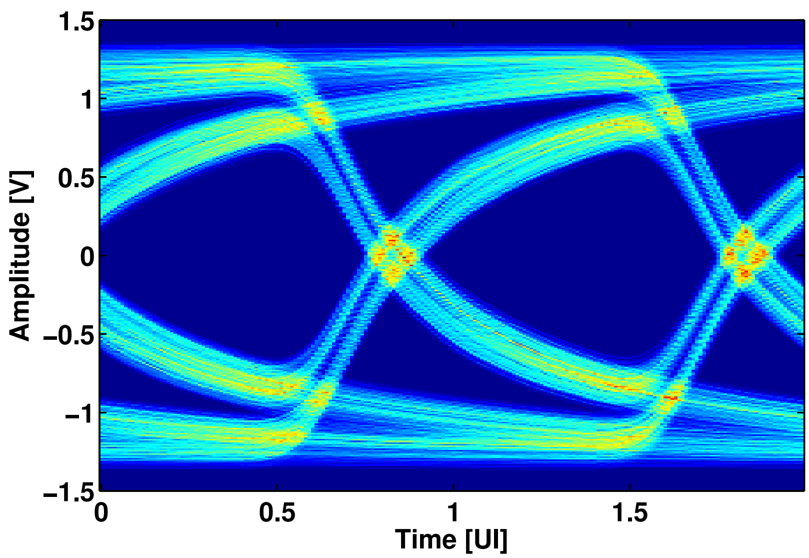


Figure 5.3: Eye diagram at 6.4 Gb/s data rate for a 29.8″ long Megtron-6 channel [82].

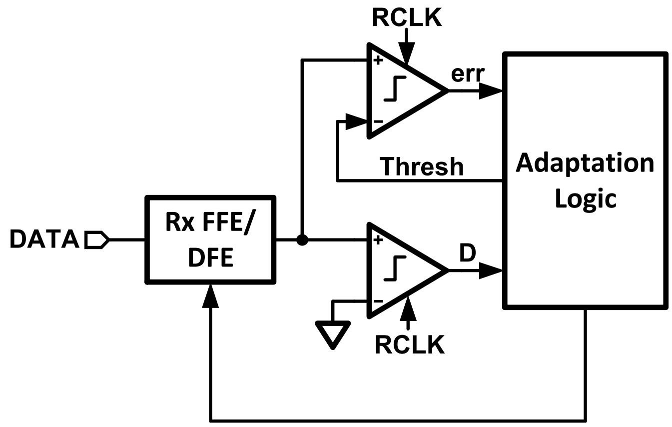


Figure 5.4: Typical adaptive equalization scheme used in high speed wireline I/O receivers [88].

5.1.2 Adaptive Equalization for High Speed Wireline I/O Interfaces

For high speed wireline I/O interfaces that operate over high loss channels, adaptive equalization of receiver feed-forward equalizer (FFE) and decision feedback equalizer (DFE) is a necessity. Use of an analog-to-digital converter (ADC) is too expensive in terms of power and complexity for high speed I/O interfaces, therefore sign-sign least mean squares (SS-LMS) algorithm is commonly used for equalizer adaptation [88]. Such scheme typically consists of an additional variable threshold sampler and adaptation logic as shown in Fig. 5.4. The variable threshold sampler provides the sign of the error signal (err) to the SS-LMS adaptation logic. Variable threshold is used when amplitude of the incoming signal is unknown and its value can be adapted using SS-LMS algorithm [88]. Other adaptation algorithms include BER-based [91], eye diagram-based [92] and spectrum-based [93]. A pulse response based method is also proposed as first step of adaptive equalization in [94]. It requires foreground calibration for pulse response estimation.

5.2 Least Mean Square Channel Estimation

The least mean square (LMS) algorithm is the most commonly used method for adaptive filtering. Figure 5.5 shows an adaptive filter used as a channel estimator [95]. The input symbols $c[k]$ are passed through a continuous time filter $h(t)$ that incorporates pulse shaping

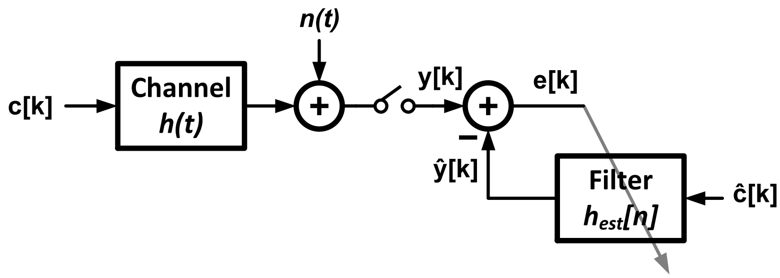


Figure 5.5: Channel estimator as adaptive filter.

filter at the transmitter as well as the actual channel response. The noise $n(t)$, assumed to be white Gaussian, is added to the continuous channel output. This noisy output is sampled by the receiver. The sampled value at the receiver can be expressed as

$$y[k] = \sum_{i=-\infty}^{\infty} c[k-i]h(t_0 + iT) + n[k] \tag{5.1}$$

where $n[k]$ is the sampled value of $n(t)$, $h(t)$ is the pulse response of the channel and t_0 is the sampling time offset. Output of an equivalent discrete time filter that mimics the channel can be expressed as

$$\hat{y}[k] = \sum_{i=-N}^{N} \hat{c}[k-i]h_{\text{est}}[i] \tag{5.2}$$

where $\hat{c}[k]$ are either estimated symbols or training symbols. The LMS algorithm for updating the filter coefficients can be written as

$$\mathbf{h_{est}}[k+1] = \mathbf{h_{est}}[k] + \mu e[k]\hat{\mathbf{c}}[k] \tag{5.3}$$

To implement the LMS algorithm in this form would require either a very high speed ADC to digitize $y[k]$ or a very high speed digital-to-analog converter (DAC) to convert $\hat{y}[k]$ to its equivalent analog voltage. Both these approaches are power hungry and result in significant overhead.

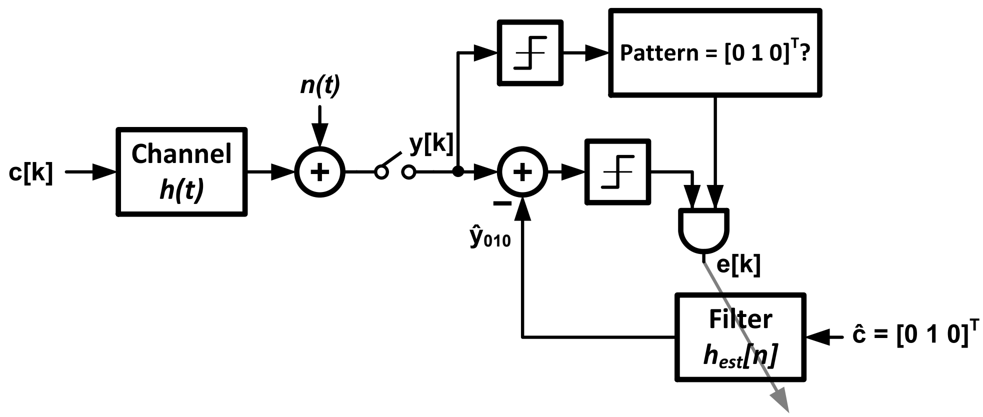


Figure 5.6: Illustration of simplified channel estimator.

5.2.1 Low Complexity LMS Estimator Implementation

The slow varying nature of wireline channels can be exploited to simplify the implementation of the LMS channel estimator. One such way would be to use a slow speed DAC corresponding to a fixed input and use error signal to update the coefficients only when the input matches the fixed input of the estimator. To illustrate this, consider a simplified channel estimation scheme shown in Fig. 5.6. Assuming that the estimator is a 3-tap finite impulse response (FIR) filter, its output for a fixed input $\hat{\mathbf{c}} = [0\ 1\ 0]^T$ is given by

$$\hat{y}_{010} = -h_{\text{est}}[-1] + h_{\text{est}}[0] - h_{\text{est}}[1] \tag{5.4}$$

If we subtract $\hat{y}_{010}$ from input $y[k]$, the resulting error would be a valid value only when $y[k]$ corresponds to input $\mathbf{c}[\mathbf{k}] = [0\ 1\ 0]^T$. By using such pattern filtering, it is possible to implement the LMS algorithm with much lower analog complexity. Ideally, we should consider all 2^N input patterns for an N-tap FIR estimator with binary inputs. If we assume the channel to be symmetric, the number of input patterns can be reduced to 2^{N-1}.

Figure 5.7 shows the detailed flow chart for channel estimation. Before beginning the estimation, we need to wait for the CDR to acquire lock. Once the CDR is locked, its output can be reliably used for pattern matching and no separate training sequence is needed for channel estimation. After the CDR is locked, estimator is initialized with coefficient values and an initial input sequence is chosen. Output of the estimator is computed for this sequence. A DAC is used to generate corresponding analog threshold voltage for error sampler. Note that the speed of the DAC is not critical as its input changes slowly and the LMS adaptation loop can discard the error samples during DAC settling time. After the DAC is settled, error samples corresponding to input pattern that matches the selected pattern are accumulated. A fixed number (N_V) of samples are accumulated for each pattern. This assumes that all the patterns are equally probable. After accumulating N_V error samples, the estimator coefficients are updated according to the LMS algorithm given by Eq. 5.3. Subsequently, the next pattern is chosen and the above procedure is repeated. After exhausting all the patterns, the estimator starts with the initial pattern and repeats this procedure as long as necessary.

The above procedure is implemented in MATLAB using floating point implementation for the channel whose pulse response at 6.4 Gb/s data rate is shown in Fig. 5.2. A signal-to-noise ratio (SNR) of 30 dB is assumed along with uniformly distributed phase jitter of 0.1 $\mathrm{UI_{pp}}$ at data rate of 6.4 Gb/s. The channel is modeled as a 3-tap FIR filter. The LMS update uses $\mu = 0.001$ and $N_V = 100$. The corresponding plot of estimated coefficient settling is shown in Fig. 5.8 for a single simulation. Figure 5.9 overlays the estimated coefficients on the channel pulse response, indicating good estimate of the pulse response at the sampling phase. Fixed point implementation for a 6-bit DAC is simulated in MATLAB and the resulting coefficient settling characteristic is shown in Fig. 5.10. An effective $\mu = 1/128$ and $N_V = 32$ are used for this simulation. The estimated coefficient values are in accordance with the channel pulse response as seen from Fig. 5.9.

5.3 Hardware Implementation

A possible implementation of the previously explained channel estimation method is shown in Fig. 5.11. The high speed input data is sampled using three samplers that output data (D), edge (E), and error (Err) samples. To be able to synthesize the digital CDR and LMS estimation logic blocks, the sampler outputs are converted to 16-bit parallel buses using 1:16 deserializers (DeSer). The digital CDR block derives 6-bit de-skew code (PI_CTRL) that is input to a phase interpolator (PI). The error sampler clock is derived from another

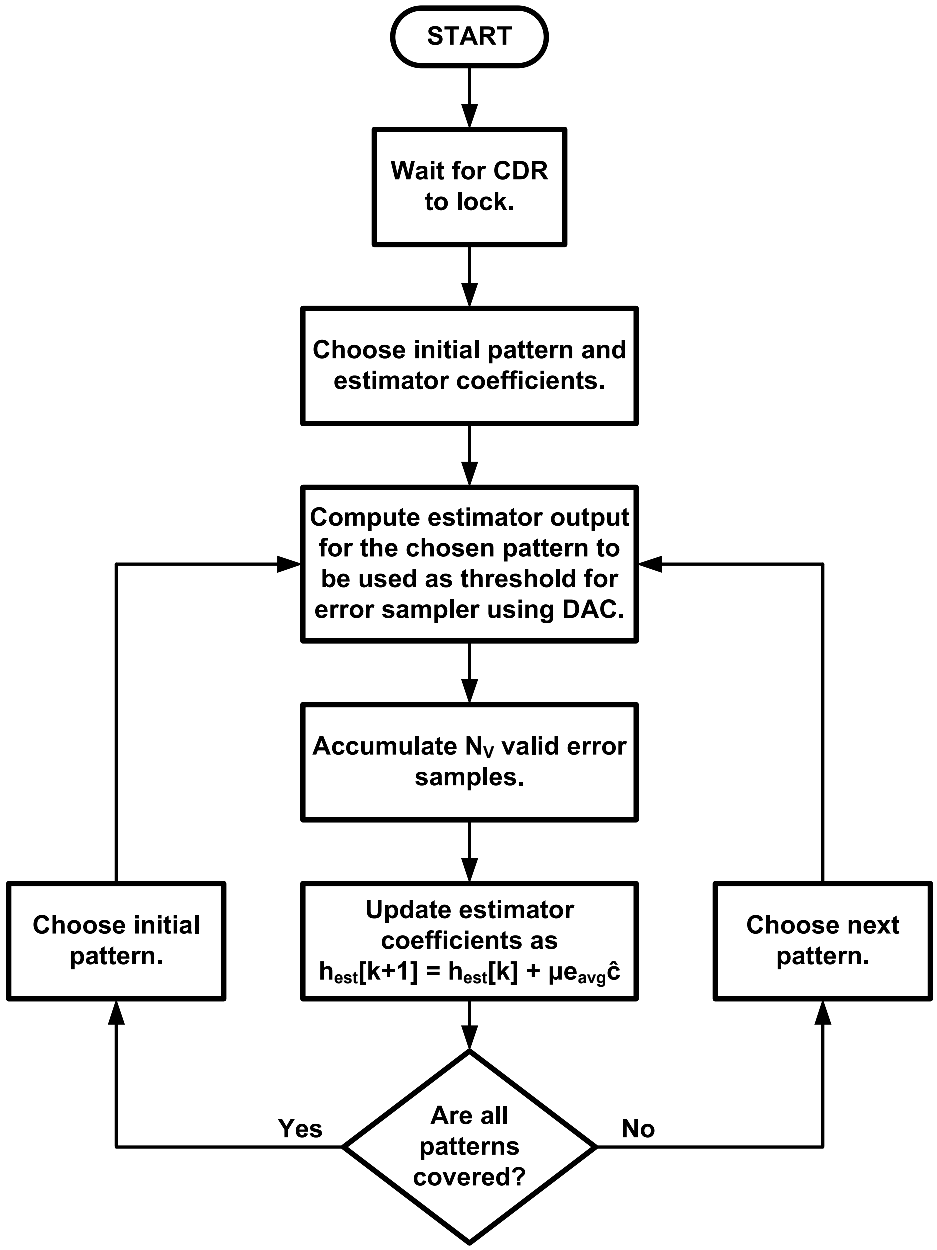


Figure 5.7: Flow chart for channel estimation.

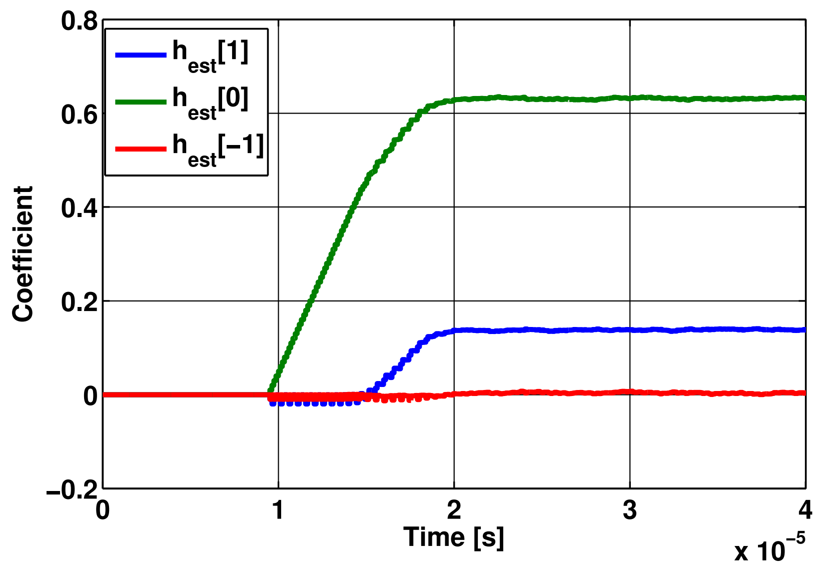


Figure 5.8: Settling characteristic of 3-tap FIR estimation filter using floating point implementation in MATLAB.

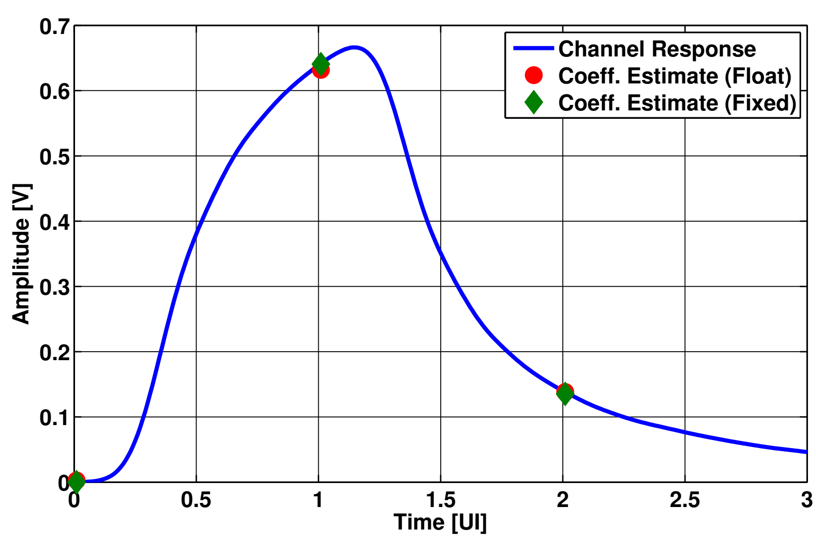


Figure 5.9: Channel pulse response and coefficients using floating point and fixed point implementations in MATLAB.

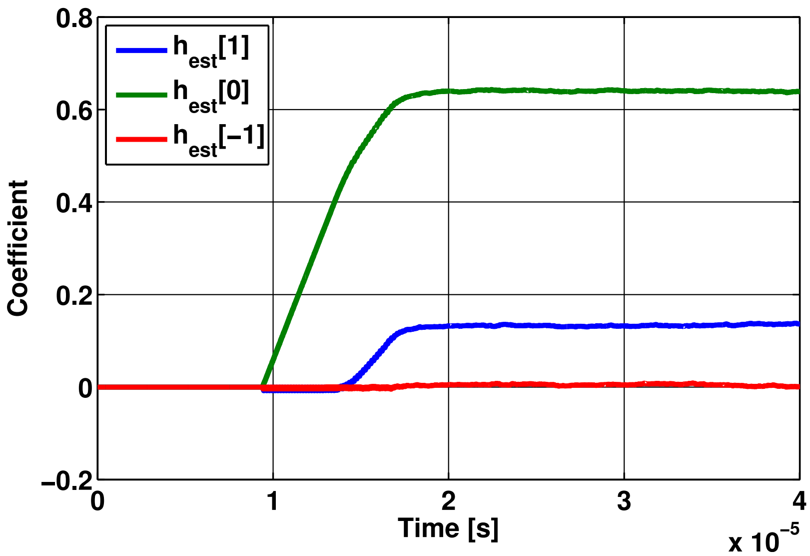


Figure 5.10: Settling characteristic of 3-tap FIR estimation filter using fixed point implementation in MATLAB.

phase interpolator that can add a fixed phase offset to the de-skew settings generated by digital CDR. By varying the phase offset, we can estimate the channel pulse response at different timing phase offsets. The LMS logic block uses data and error samples to derive 7-bit code for DAC. The DAC output is used as threshold voltage of the error sampler. For the simulations, behavioral models are used for all the blocks except the digital CDR block and LMS logic block. Synthesizable Verilog code is written for digital CDR as well as LMS logic block.

The simplified implementation of the adaptive estimation filter is shown in Fig. 5.12. Note that the delays are shown z^{-1} for convenience. The update rate of the filter is not uniform by design. Taking advantage of binary nature of the input, no multipliers are used for implementation. Further, both step size (μ) scaling as well as truncation operations are achieved at the output of the filter by dropping 6 LSBs of the adder output. Two's complement arithmetic is used everywhere. The filter uses accumulated valid error samples to generate 7-bit DAC output code.

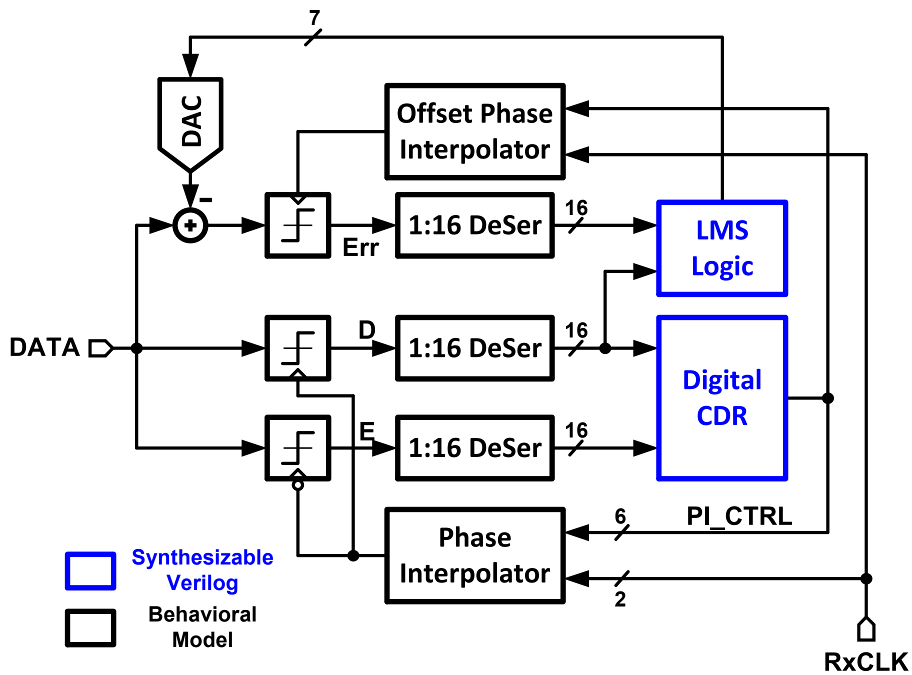


Figure 5.11: Block diagram of LMS channel estimation implementation.

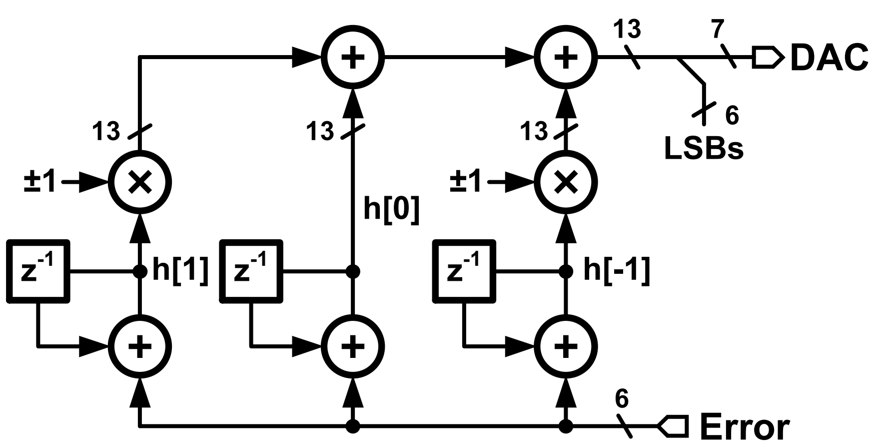


Figure 5.12: Simplified adaptive estimation filter.

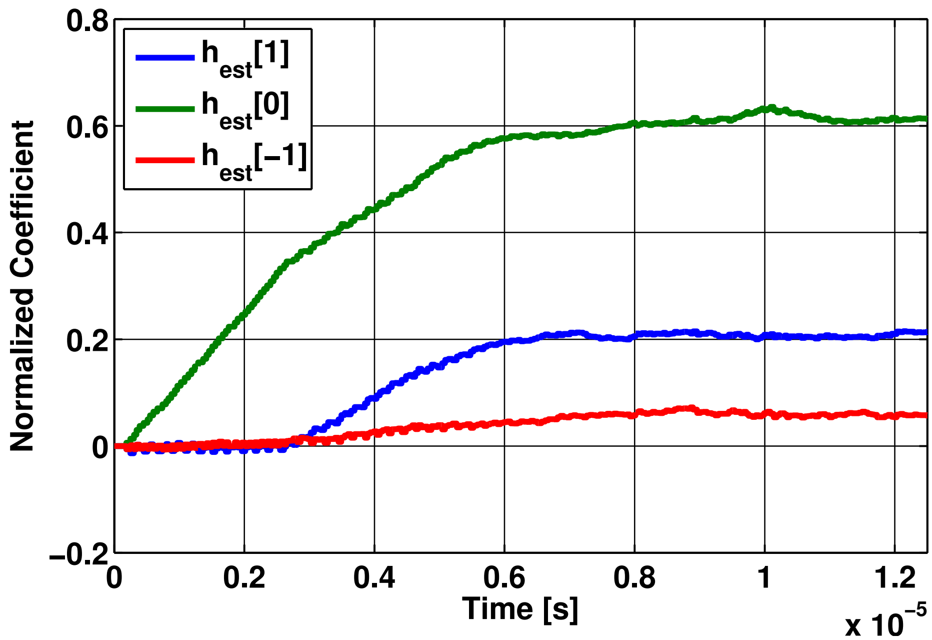


Figure 5.13: Settling characteristic of 3-tap FIR estimation filter using AMS simulation.

5.3.1 Simulation Results

Channel estimation block as well as digital CDR are simulated with the channel shown in Fig. 5.1 at 6.4 Gb/s data rate using Verilog AMS. An ideal transmitter with clock jitter of 1 $\mathrm{ps_{rms}}$ is assumed. Maximum transmitter amplitude is $250\mathrm{mV_{pk,diff}}$ whereas DAC full scale is assumed to be $\pm 275\mathrm{mV_{diff}}$. Figure 5.13 shows the simulated settling characteristic of the LMS channel estimation filter. The coefficient values are normalized to their full scale values which in turn depend on the DAC full scale.

By noting that

$$h_{\text{est}}[i] = h(t_0 + iT) \tag{5.5}$$

when error samples are taken at offset, t_0, we can reconstruct the complete pulse response by sweeping the offset t_0 of the error sample. In hardware implementation, this is achieved by using a separate offset PI for generating error samples. For a 3-tap FIR filter based

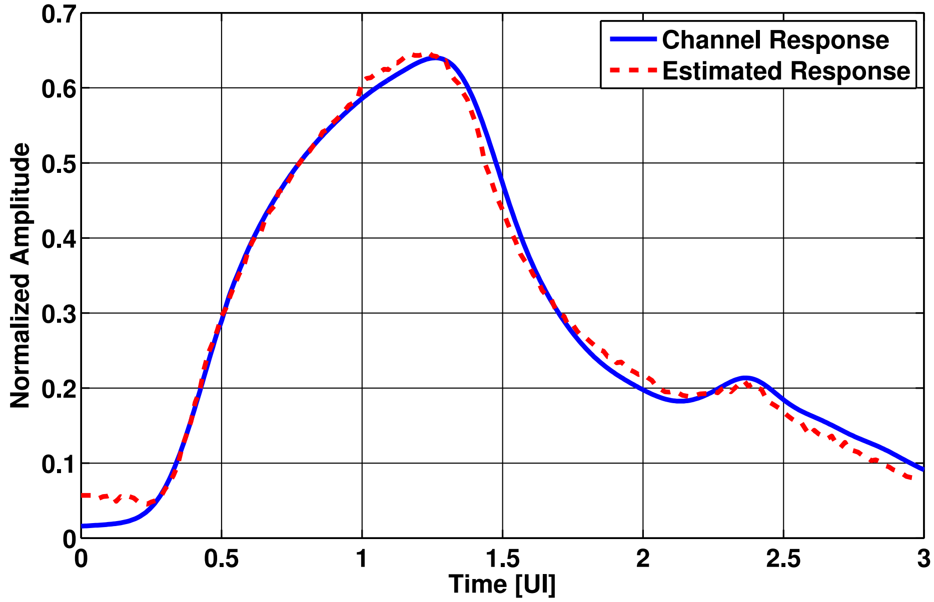


Figure 5.14: Simulated and estimated pulse response of the channel at data rate of 6.4 Gb/s.

estimation, pulse response can be estimated for 3 UI length. The resulting estimated pulse response along with the simulated channel pulse response is shown in Fig. 5.14. Good agreement between simulated pulse response and estimated pulse response is observed.

The digital CDR and the estimator block are synthesized using a 65 nm CMOS process using regular V_t transistors. The area and power of these blocks after synthesis are shown in Table 5.1. Note that the power numbers are estimated for 0.9 V supply voltage and slow corner at 125° C temperature. These metrics indicate the low overhead of the digital blocks. Due to relaxed speed requirement of the DAC, it is also expected to consume less power. In such case most of the power will be consumed by the deserializer for error samples.

Table 5.1: Post-synthesis area and power consumption of Verilog blocks.

Technology	65 nm CMOS
Clock Frequency	400 MHz
No. of Comb. Cells	714
No. of Seq. Cells	159
Total Cell Area	4813 μm^2
Cell Internal Power	0.548 mW
Net Switching Power	0.351 mW
Leakage Power	0.041 mW
Total Power	0.94 mW

5.4 Conclusion

We have presented a low complexity channel pulse response estimator using LMS algorithm in this chapter. MATLAB as well as hardware implementation-based simulations were performed to verify the functionality of the estimator. Pulse response is an important characteristic of the channel. In future, it is possible to use the estimated pulse response to adaptively equalize the channel. Presently LMS techniques cannot be directly applied for combined adaptation of continuous-time equalizer such as CTLE and front-end variable gain amplifier (VGA). Estimated channel pulse response can be utilized to adapt CTLE and VGA together.

CHAPTER 6

DESIGN OF A TWO-STAGE DIGITAL FRACTIONAL-N PLL

Highly digital architectures for fractional-N PLLs have recently gained popularity due to their portability, reconfigurability, and compatibility with manufacturing processes optimized for digital circuits. Use of digital loop filter in fractional-N PLLs obviates the need for external loop filter components, providing area and cost benefits over their analog counterparts. To leverage these benefits, various digital fractional-N PLL architectures have been proposed. These can be broadly classified in following four categories: (i) PLLs with integer $\Delta\Sigma$ dividers and time-to-digital converters (TDCs) [96, 97], (ii) high resolution fractional divider-based PLLs [98, 99, 100], (iii) fractional counter-based PLLs [101, 102], and (iv) $\Delta\Sigma$ frequency-to-digital converter-based PLLs [103, 104, 105].

An integer $\Delta\Sigma$ divider and TDC-based fractional-N PLL is shown in Fig. 6.1(a). It resembles closely to a conventional analog $\Delta\Sigma$ fractional-N PLL. It consists of a time-to-digital converter (TDC) that detects the phase difference between reference input (REF) and the output of multi-modulus divider (MMD). The output of TDC is processed by a digital loop filter (DLF), whose output controls the digitally controlled oscillator (DCO). The division ratio of the multi-modulus divider is dithered using a digital $\Delta\Sigma$ modulator according to the fractional control word N_{FRAC}. The TDC replaces charge pump and phase detector in the analog PLL. The digital loop filter replaces the analog loop filter. The digital fractional-N PLL occupies much smaller area compared to its analog counterpart thanks to the digital loop filter. The digital architecture is also more friendly to process scaling. Additionally, digital PLL architecture is more amenable to quantization noise cancellation. The input phase difference seen by the TDC consists of a deterministic component and a random component. The deterministic component is caused by the use of $\Delta\Sigma$ modulator to control MMD and depends on the difference between fractional control word N_{FRAC} and

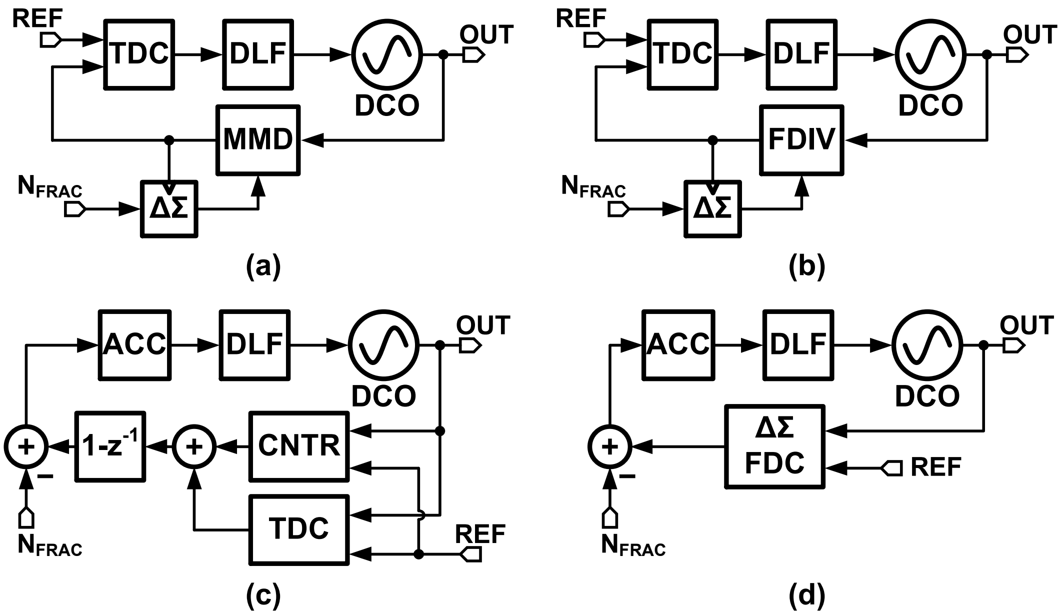


Figure 6.1: Simplified block diagrams of (a) integer $\Delta\Sigma$ divider and TDC-based PLL, (b) fractional $\Delta\Sigma$ divider-based PLL, (c) fractional counter-based PLL, and (d) $\Delta\Sigma$ FDC-based PLL.

digital $\Delta\Sigma$ modulator output. This difference can then be subtracted from TDC output to cancel the deterministic quantization noise of the digital $\Delta\Sigma$ modulator. Such quantization noise cancellation scheme enables wide bandwidth operation of the fractional-N PLL. For the cancellation to be effective, a high resolution TDC as well as precise knowledge of TDC gain is necessary. TDC gain calibration mechanisms such as least-mean squares adaptation loop [96] increase the PLL design complexity.

The second type of PLL architecture, shown in Fig. 6.1(b), uses a high resolution fractional divider (FDIV) to significantly reduce the quantization noise generated due to the $\Delta\Sigma$ divider. This relaxes the resolution requirement of the TDC, enabling use of a 1-bit bang-bang phase detector [98]. The fractional divider can be implemented using a delay-chain based digital-to-time converter (DTC) [98] or a phase interpolator [99]. Delay-chain based DTC needs to be calibrated to match its digital-to-delay gain to one DCO period, resulting in increased complexity. On the other hand, phase interpolator-based DTC does not require gain calibration [99].

A simplified block diagram of a fractional counter based PLL is depicted in Fig. 6.1(c). The combination of integer counter (CNTR), TDC, and differentiator block $(1-z^{-1})$ operates as a fractional counter that provides the number of output cycles in one reference period with a fractional precision. The frequency control word N_{FRAC} is subtracted from the count to obtain DCO frequency error. An accumulator block (ACC) accumulates the DCO frequency error to provide an estimate of phase error to the digital loop filter, DLF. The precision of the fractional counter is decided by the TDC precision. Furthermore, the TDC gain also needs to be calibrated so that its full scale output corresponds to input phase error of one DCO period [101].

It is interesting to note that the fractional counter can be thought of as a flash frequency-to-digital converter (FDC). The FDC-based PLL (FDCPLL) shown in Fig. 6.1(d) instead utilizes an oversampled $\Delta\Sigma$ frequency-to-digital converter ($\Delta\Sigma$ FDC) [103, 104, 105]. Fractional frequency control word N_{FRAC} is subtracted from $\Delta\Sigma$ FDC output to estimate DCO frequency error. This error is accumulated using an accumulator (ACC) to estimate phase error. The digital loop filter processes this error to control DCO frequency. FDCPLL offers advantages similar to TDC-based PLLs such as low area and scaling friendly nature due to their highly digital implementation. The $\Delta\Sigma$ FDC exploits noise-shaping and oversampling to improve its accuracy. Therefore, unlike the fractional counter based PLL architecture, FDCPLL does not require a high resolution TDC. To implement quantization noise cancellation, knowledge of quantizer gain inside the $\Delta\Sigma$ FDC is necessary. Typically the multi-bit quantizer gain is fixed by design [106, 105] to enable quantization noise cancellation.

Most implementations of the previously described architectures require calibration of TDC or DTC gain to improve their performance. Additionally, a high resolution TDC or DTC is also needed for effective quantization noise cancellation. In this chapter, we present a PLL architecture that avoids use of high resolution TDC or DTC and does not require any gain calibration mechanism for low power, low jitter operation. This is achieved by using a fractional divider based 1-bit first order $\Delta\Sigma$ FDC. Use of a phase interpolator (PI) for fractional division obviates the need for calibration while first order $\Delta\Sigma$ FDC enables a simple implementation. We also propose the use of multiplying delay-locked loop (MDLL) based integer-N reference multiplication to exploit the benefits of oversampling in a $\Delta\Sigma$ FDC. We discuss 1-bit first order $\Delta\Sigma$ FDC in more detail in the next section. In Section 6.2 we provide details of the proposed two-stage PLL architecture. The implementation of critical circuits is described in Section 6.3. The measurement results are shown in Section 6.4. We conclude the chapter by summarizing the findings of this work in Section 6.5.

6.1 $\Delta\Sigma$ Frequency to Digital Converter

Principles of oversampling and noise-shaping are often used in analog-to-digital and digital-to-analog conversion systems. $\Delta\Sigma$ FDCs utilize these same principles for high resolution frequency-to-digital conversion while utilizing coarse quantizers. A $\Delta\Sigma$ FDC with DCO in the feedback was proposed in [107], while a $\Delta\Sigma$ FDC utilizing a divider in the feedback was proposed in [108]. As depicted in Fig. 6.2(a) a basic 1-bit first order $\Delta\Sigma$ FDC consists of a dual-modulus divider (DMD) controlled by the output of a D flip-flop (DFF) that acts as a 1-bit phase quantizer (PQ) [108]. A high frequency clock, $\mathrm{CK_{DCO}}$, is divided by a factor of N or N+1 based on the PQ output, $\mathrm{D_{OUT}}$. The output of DMD is used as D input of the phase quantizer flip-flop. Reference clock input ($\mathrm{CK_{REF}}$) is used as sampling clock of the phase quantizer flip-flop. Illustrative steady-state waveforms when frequency of $\mathrm{CK_{DCO}}$ ($\mathrm{F_{DCO}}$) is 2.25 times frequency of $\mathrm{CK_{REF}}$ ($\mathrm{F_{REF}}$) and N=2 are shown in Fig. 6.2(b). In this case when $\mathrm{CK_{REF}}$ is lagging the divider output $\mathrm{CK_{DIV}}$, the division ratio is changed to 3; otherwise the division ratio is equal to 2. In steady state the FDC loop operates such that the average phase difference between $\mathrm{CK_{DIV}}$ and $\mathrm{CK_{REF}}$ is zero. An equivalent model of the basic first order $\Delta\Sigma$ FDC is shown in Fig. 6.3. Let $\tau_{\mathrm{REF}}[n]$ and $\tau_{\mathrm{DCO}}[n]$ be the period sequences of the reference clock ($\mathrm{CK_{REF}}$) and DCO output ($\mathrm{CK_{DCO}}$), respectively. The zero time crossing sequence of $\mathrm{CK_{REF}}$ is denoted as $t_{\mathrm{REF}}[n]$ and zero crossing time sequence of the divider output ($\mathrm{CK_{DIV}}$) is denoted as $t_{\mathrm{DIV}}[n]$. The transformation of period sequence to zero crossing time sequence corresponds to an implicit integration from frequency to phase. Therefore the DMD combines the functions of a DAC as well as integrator in $\Delta\Sigma$ FDC. The D flip-flop plays the role of a 1-bit phase quantizer (PQ). The 1-bit $\Delta\Sigma$ FDC is analogous to a 1-bit $\Delta\Sigma$ ADC and it digitizes the period difference, or equivalently the frequency difference, between its inputs, $\mathrm{CK_{REF}}$ and $\mathrm{CK_{DCO}}$. It can be shown that the basic first order

$\Delta\Sigma$ FDC satisfies the following equation [108]:

$$\Delta t[n] = \Delta t[n-1] + \{(N+0.5)\tau_{\mathrm{DCO}}[n-1] - \tau_{\mathrm{REF}}[n-1]\} - 0.5sgn(\Delta t[n-1])\tau_{\mathrm{DCO}}[n-1] \quad (6.1)$$

where $\Delta t[n] = t_{\mathrm{REF}}[n] - t_{\mathrm{DIV}}[n]$ and $sgn(x)$ are the sequence of input phase error to the quantizer and sign function, respectively. In steady state, $E\{\Delta t[n]\} = E\{\Delta t[n-1]\}$, where $E\{\cdot\}$ denotes the mean value operator. This results in

$$\mathrm{D}_{\mathrm{OUT,avg}} = \frac{\tau_{\mathrm{REF,avg}} - (N+0.5)\tau_{\mathrm{DCO,avg}}}{\tau_{\mathrm{DCO,avg}}} \quad (6.2)$$

where

$$\mathrm{D}_{\mathrm{OUT}}[n] = sgn(\Delta t[n])$$

is the digital output of the $\Delta\Sigma$ FDC. $\tau_{\mathrm{REF,avg}}$ and $\tau_{\mathrm{DCO,avg}}$ denote the average period of the $\mathrm{CK}_{\mathrm{REF}}$ and $\mathrm{CK}_{\mathrm{DCO}}$, respectively. In terms of frequency, we can also write

$$\mathrm{D}_{\mathrm{OUT,avg}} = \frac{\mathrm{F}_{\mathrm{DCO,avg}}}{\mathrm{F}_{\mathrm{REF,avg}}} - (N+0.5) \quad (6.3)$$

Figure 6.4(a) depicts how a $\Delta\Sigma$ FDC may be used to achieve fractional-N frequency multiplication [104, 105]. The DCO output clock ($\mathrm{CK}_{\mathrm{DCO}}$) as well as reference clock ($\mathrm{CK}_{\mathrm{REF}}$) are fed to $\Delta\Sigma$ FDC. In steady state, the output of the $\Delta\Sigma$ FDC equals the fractional frequency difference, α, between $\mathrm{CK}_{\mathrm{DCO}}$ and $\mathrm{CK}_{\mathrm{REF}}$. Frequency error signal, $\mathrm{F}_{\mathrm{ERR}}$, is obtained by subtracting this fractional offset $\alpha \in (-0.5, 0.5)$ from $\Delta\Sigma$ FDC output. A digital accumulator (ACC) accumulates $\mathrm{F}_{\mathrm{ERR}}$ to estimate the phase error signal, Φ_{ERR}. A proportional-integral control-based digital loop filter (DLF) processes the phase error and controls the DCO. By virtue of DLF integral path, in steady-state it is ensured that

$$\begin{aligned} \tau_{\mathrm{REF,avg}} &= (N+0.5+\alpha)\tau_{\mathrm{DCO,avg}} \\ \text{i.e.} \quad \mathrm{F}_{\mathrm{DCO,avg}} &= (N+0.5+\alpha)\mathrm{F}_{\mathrm{REF,avg}} \end{aligned}$$

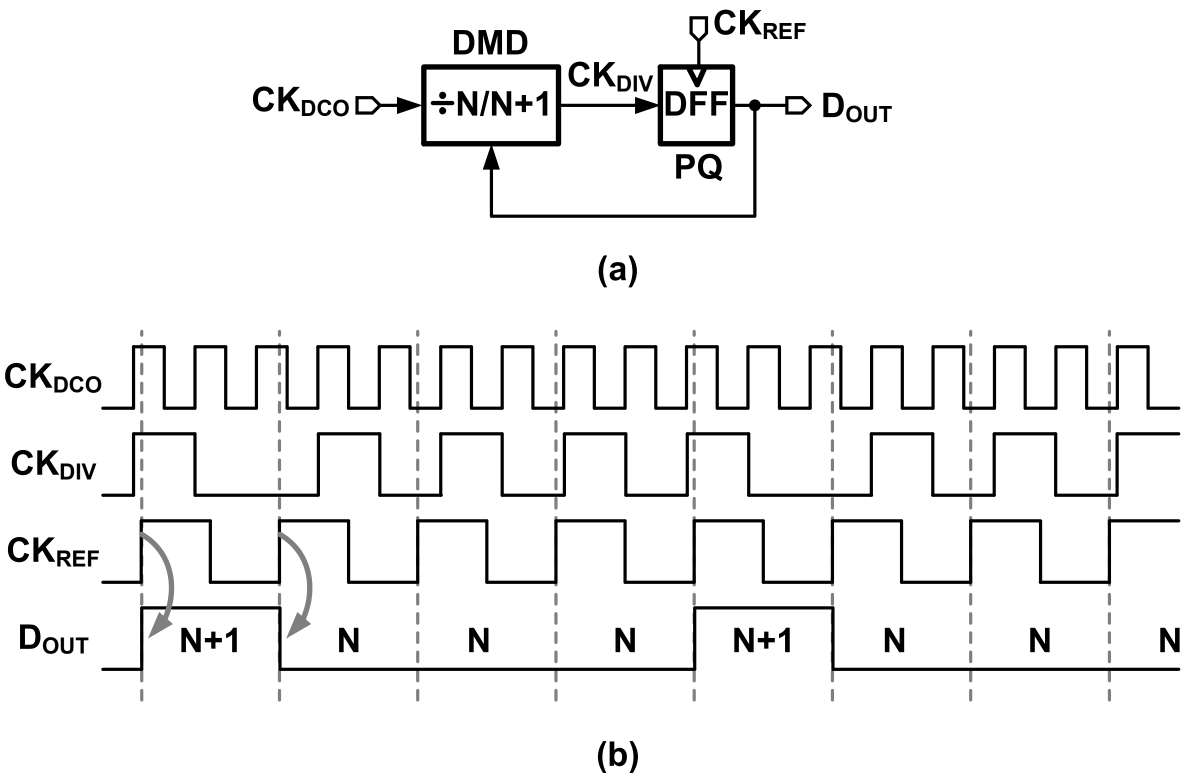


Figure 6.2: (a) Block diagram of a basic 1-bit first order $\Delta\Sigma$ frequency-to-digital converter, and (b) illustrative waveforms for $F_{DCO} = 2.25F_{REF}$, N=2.

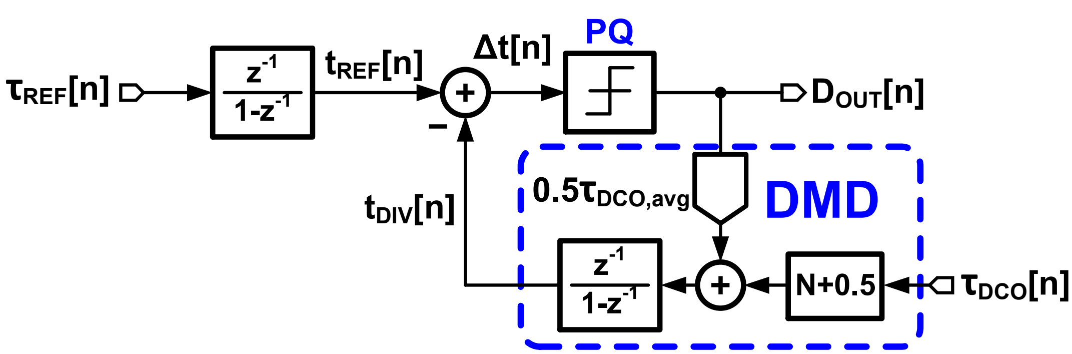
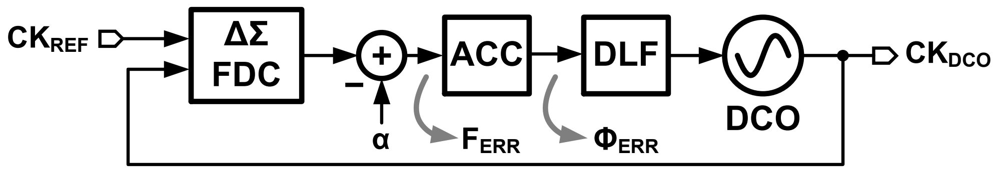


Figure 6.3: Block diagram of a basic 1-bit first order $\Delta\Sigma$ frequency-to-digital converter.



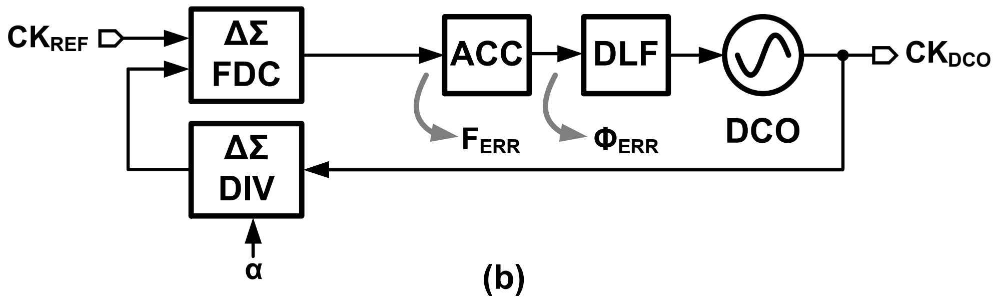


Figure 6.4: Simplified block diagram of (a) FDCPLL with non-zero ΔΣ FDC output, and (b) FDCPLL with zero ΔΣ FDC output.

An alternate method of using a ΔΣ FDC in a fractional-N PLL is shown in Fig. 6.4(b) [109, 110]. A digital ΔΣ modulator based integer divider (ΔΣ DIV) is used to provide a divided clock input to the ΔΣ FDC. This is analogous to an analog ΔΣ fractional-N PLL. Due to the ΔΣ integer divider, the average input to the ΔΣ FDC is always zero. As a result no separate subtraction of the fractional offset is necessary. As before, the output of ΔΣ FDC denotes the frequency error F_{ERR}. A phase accumulator integrates F_{ERR} to obtain phase error estimate Φ_{ERR}, which is processed by digital loop filter to control DCO.

In the context of first order ΔΣ FDC with 1-bit quantizer, the FDCPLL architecture shown in Fig. 6.4(b) offers some interesting advantages. Fixing average input of the ΔΣ FDC nominally to zero improves its tonal behavior. Furthermore, a technique for feedback gain scaling using the digital ΔΣ modulator of the ΔΣ DIV can also be applied to increase the in-band gain of the ΔΣ FDC [109]. Note that this architecture for ΔΣ FDC with zero average input is called phase minimization loop (PML) in [109]. In the next sub-section we delve into the details of zero input ΔΣ FDC.

6.1.1 Zero Input $\Delta\Sigma$ FDC

Figure 6.5(a) depicts a basic zero input first order $\Delta\Sigma$ FDC. The output of the phase quantizer (PQ) is added to the output of a digital $\Delta\Sigma$ modulator ($\Delta\Sigma$) to generate the multi-modulus divider (MMD) modulus $v[n]$. It is assumed that the phase quantizer (PQ) output is either 0 or 1. In steady state the mean value of $v[n]$ is

$$\begin{aligned} v_{\mathrm{avg}} &= \alpha + \mathrm{D}_{\mathrm{OUT,avg}} \\ \mathrm{D}_{\mathrm{OUT,avg}} &= \frac{\tau_{\mathrm{REF,avg}} - (N + v_{\mathrm{avg}})\tau_{\mathrm{DCO,avg}}}{\tau_{\mathrm{DCO,avg}}} \end{aligned} \tag{6.4}$$

where N is the integer part of the multiplication factor. When used in a FDC-based PLL, we can force the average $\Delta\Sigma$ FDC output to mid-scale (i.e. 0.5 in this case), which results in

$$\begin{aligned} \tau_{\mathrm{REF,avg}} &= (N + 0.5 + \alpha)\tau_{\mathrm{DCO,avg}} \\ \text{i.e.}\quad \mathrm{F}_{\mathrm{DCO,avg}} &= (N + 0.5 + \alpha)\mathrm{F}_{\mathrm{REF,avg}} \end{aligned}$$

Figure 6.5(b) shows another zero input first order $\Delta\Sigma$ FDC that uses gain-scaled feedback [109]. The output of phase quantizer (PQ) is scaled by a factor $\mathrm{K}_{\mathrm{FB}} < 1$ and added to the input of a digital $\Delta\Sigma$ modulator ($\Delta\Sigma$) to generate the multi-modulus divider (MMD) modulus $v[n]$. It is assumed that the phase quantizer (PQ) output is either -1 or +1. In steady-state the mean value of $v[n]$ is

$$\begin{aligned} v_{\mathrm{avg}} &= \alpha + \mathrm{K}_{\mathrm{FB}}\mathrm{D}_{\mathrm{OUT,avg}} \\ \mathrm{D}_{\mathrm{OUT,avg}} &= \left(\frac{\tau_{\mathrm{REF,avg}} - (N + v_{\mathrm{avg}})\tau_{\mathrm{DCO,avg}}}{\tau_{\mathrm{DCO,avg}}}\right) \end{aligned}$$

When used in a FDCPLL, we can force the mean $\Delta\Sigma$ FDC output to mid-scale (i.e. 0 in

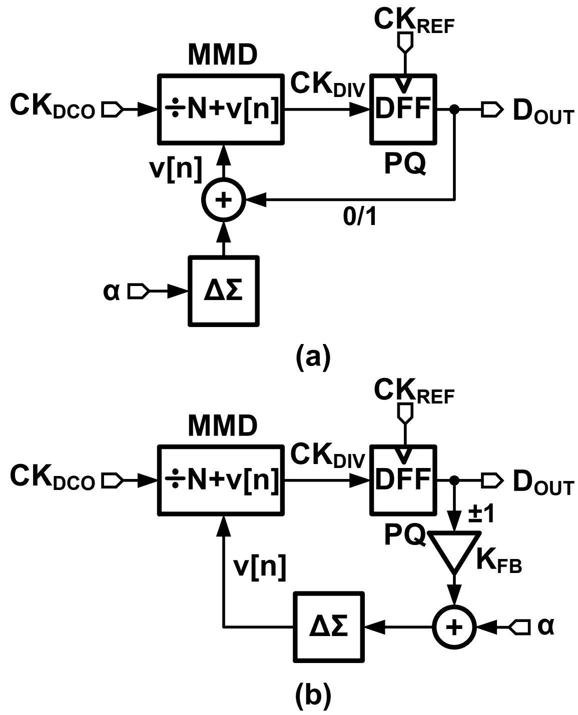


Figure 6.5: Simplified block diagram of (a) basic zero input $\Delta\Sigma$ FDC, and (b) zero input $\Delta\Sigma$ FDC with feedback gain scaling [109].

this case), which results in

$$\begin{aligned} \tau_{\mathrm{REF,avg}} &= (N+\alpha)\tau_{\mathrm{DCO,avg}} \\ \text{i.e.}\quad \mathrm{F}_{\mathrm{DCO,avg}} &= (N+\alpha)\mathrm{F}_{\mathrm{REF,avg}} \end{aligned}$$

It is important to note that the use of digital $\Delta\Sigma$ modulator enables use of $\mathrm{K}_{\mathrm{FB}} < 2^{-1}$ which is not possible to do otherwise.

Having looked at various first order $\Delta\Sigma$ FDC architectures, next we describe their impact on FDCPLL performance.

6.1.2 TDC Analogy for $\Delta\Sigma$ FDC

To study the impact of $\Delta\Sigma$ FDC on PLL performance, it is useful to observe that the cascade of $\Delta\Sigma$ FDC and digital accumulator is analogous to a high resolution TDC as depicted in Fig. 6.6. The first order differences of reference phase ($\Phi_{\mathrm{REF}}(z)$) and DCO phase ($\Phi_{\mathrm{DCO}}(z)$)

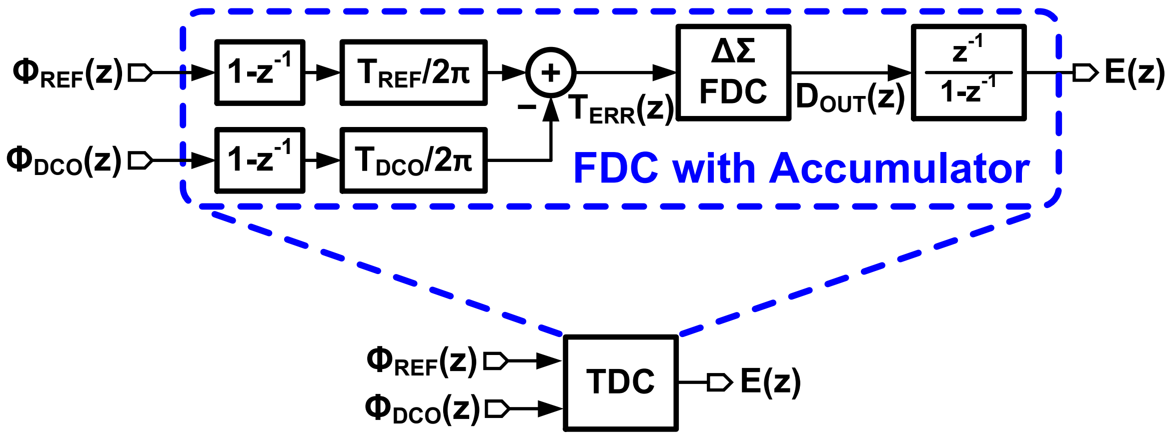


Figure 6.6: Cascade of ΔΣ FDC and accumulator viewed as an equivalent TDC.

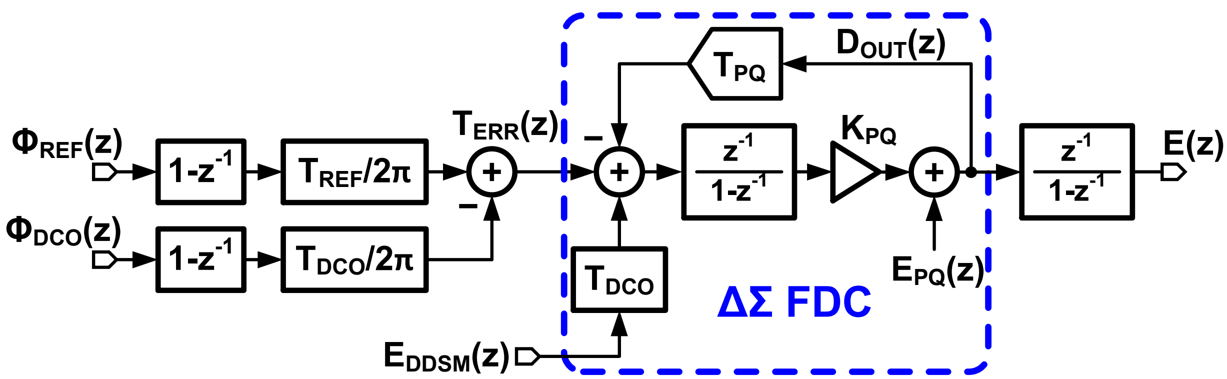


Figure 6.7: Detailed small signal model for cascade of ΔΣ FDC and accumulator.

are scaled to obtain the period error signal, $\mathrm{T_{ERR}}(z)$. $\mathrm{T_{REF}}$ and $\mathrm{T_{DCO}}$ denote the average time period of the reference clock and DCO output, respectively. The ΔΣ FDC digitizes this signal to generate its output $\mathrm{D_{OUT}}(z)$. A digital accumulator converts $\mathrm{D_{OUT}}(z)$ to digital phase error signal, $\mathrm{E}(z)$. This is equivalent to a TDC with inputs Φ_{REF} and Φ_{DCO} and output $\mathrm{E}(z)$. Therefore, by deriving equivalent TDC characteristics, we can utilize the conventional TDC-based PLL analysis techniques to predict the impact of ΔΣ FDC characteristics on FDCPLL performance.

A detailed small signal model of the ΔΣ FDC with accumulator is shown in Fig. 6.7. As described above, the period error signal, $\mathrm{T_{ERR}}(z)$, is the difference between reference period and DCO period. It is given by

$$\mathrm{T_{ERR}}(z) = (1 - z^{-1})\frac{\mathrm{T_{REF}}}{2\pi}\Phi_{\mathrm{REF}}(z) - (1 - z^{-1})\frac{\mathrm{T_{DCO}}}{2\pi}\Phi_{\mathrm{DCO}}(z) \tag{6.5}$$

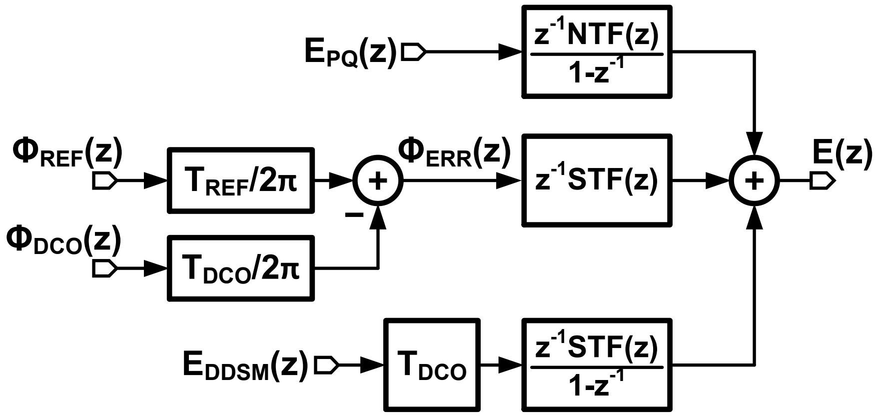


Figure 6.8: Simplified small signal model with $\Delta\Sigma$ transfer functions for cascade of $\Delta\Sigma$ FDC and accumulator.

Note that average reference period ($\mathrm{T_{REF}}$) and average DCO period ($\mathrm{T_{DCO}}$) are related by frequency multiplication factor, $\mathrm{N_{nom}}$, as

$$\mathrm{T_{REF}} = \mathrm{N_{nom}}\mathrm{T_{DCO}}$$

The quantization error added by the digital $\Delta\Sigma$ modulator is denoted as $\mathrm{E_{DDSM}}(z)$, while $\mathrm{E_{PQ}}(z)$ denotes the quantization error of the phase quantizer. The linearized gain of the 1-bit phase quantizer is denoted as $\mathrm{K_{PQ}}$, and the quantizer feedback gain is denoted as $\mathrm{T_{PQ}}$. We denote the signal transfer function (STF) of the $\Delta\Sigma$ FDC as $\mathrm{STF}(z)$ and phase quantization noise transfer function (NTF) as $\mathrm{NTF}(z)$. Therefore,

$$\begin{aligned}\mathrm{STF}(z) &\triangleq \frac{\mathrm{D_{OUT}}(z)}{\mathrm{T_{ERR}}(z)}\\ \mathrm{NTF}(z) &\triangleq \frac{\mathrm{D_{OUT}}(z)}{\mathrm{E_{PQ}}(z)}\end{aligned}$$

With the above transfer functions in mind, the simplified small signal model is shown in Fig. 6.8. Contributions of reference phase noise, DCO phase noise as well as digital $\Delta\Sigma$ quantization noise and phase quantization noise to the digital phase error output can be found if we know the $\mathrm{STF}(z)$ and $\mathrm{NTF}(z)$ of the $\Delta\Sigma$ FDC. Therefore, in the following discussion, we focus our attention on the $\Delta\Sigma$ FDC transfer functions.

To understand the benefit of feedback gain scaling, consider the simplified small signal

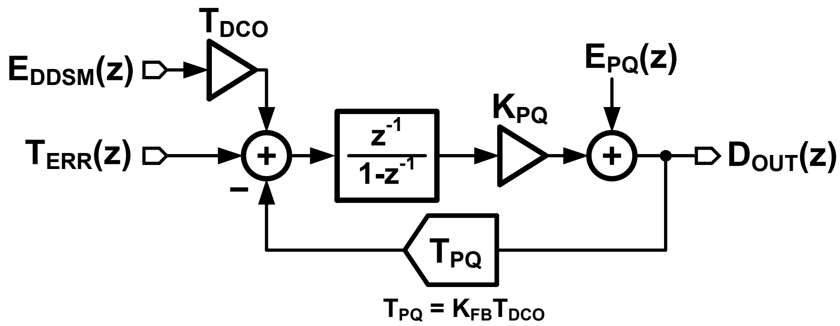


Figure 6.9: Simplified small signal equivalent block diagram for zero input ΔΣ FDC with feedback gain scaling [109].

equivalent model shown in Fig. 6.9. A constant gain $\mathrm{K_{PQ}}$ is used as a linear model for the 1-bit phase quantizer. The feedback gain, $\mathrm{T_{PQ}}$, of the phase quantizer is equal to $\mathrm{K_{FB}T_{DCO}}$. If we assume that there is no overloading of the ΔΣ FDC and that phase quantization noise $\mathrm{E_{PQ}}(z)$ is white with power spectral density of $\dfrac{2}{3\mathrm{F_{REF}}}\left(\mathrm{LSB^2/Hz}\right)$, the STF of the ΔΣ FDC is given by

$$\mathrm{STF}(z) = \frac{\mathrm{K_{PQ}}z^{-1}}{1-(1-\mathrm{K_{PQ}T_{PQ}})z^{-1}} \tag{6.6}$$

The DC gain is given by

$$\left.\frac{\mathrm{D_{OUT}}(z)}{\mathrm{T_{ERR}}(z)}\right|_{z=1} = \frac{1}{\mathrm{T_{PQ}}} = \frac{1}{\mathrm{K_{FB}T_{DCO}}} \tag{6.7}$$

Note that the DC gain of the ΔΣ FDC STF does not depend on the linearized phase quantizer gain $\mathrm{K_{PQ}}$. This property of ΔΣ FDC obviates the need for DC gain calibration of the ΔΣ FDC despite using a 1-bit phase quantizer that is typically needed in bang-bang PLLs [109]. NTF of the ΔΣ FDC is given by

$$\mathrm{NTF}(z) = \frac{1-z^{-1}}{1-(1-\mathrm{K_{PQ}T_{PQ}})z^{-1}} \tag{6.8}$$

Based on Fig. 6.8, the digital phase error due to $\Phi_{\mathrm{ERR}}(z)$ is given by

$$\frac{\mathrm{E}(z)}{\Phi_{\mathrm{ERR}}(z)} = \frac{\mathrm{K_{PQ}}z^{-2}}{1-(1-\mathrm{K_{PQ}T_{PQ}})z^{-1}} \tag{6.9}$$

Similarly, the digital phase error due to digital ΔΣ quantization noise as well as PQ quantization noise is given by

$$\begin{aligned} \mathrm{E}_{\mathrm{Q,Total}}(z) &= \mathrm{E}_{\mathrm{DDSM}}(z)\mathrm{T}_{\mathrm{DCO}}\frac{\mathrm{K}_{\mathrm{PQ}}z^{-2}}{(1-(1-\mathrm{K}_{\mathrm{PQ}}\mathrm{T}_{\mathrm{PQ}})z^{-1})(1-z^{-1})} \\ &+ \mathrm{E}_{\mathrm{PQ}}(z)\frac{z^{-1}}{1-(1-\mathrm{K}_{\mathrm{PQ}}\mathrm{T}_{\mathrm{PQ}})z^{-1}} \end{aligned} \tag{6.10}$$

For low-to-mid frequencies, the contribution from $\mathrm{E}_{\mathrm{PQ}}(z)$ is much larger than that from $\mathrm{E}_{\mathrm{DDSM}}(z)$. Therefore,

$$\mathrm{E}_{\mathrm{Q,Total}}(z) \approx \mathrm{E}_{\mathrm{PQ}}(z)\frac{z^{-1}}{1-(1-\mathrm{K}_{\mathrm{PQ}}\mathrm{T}_{\mathrm{PQ}})z^{-1}} \tag{6.11}$$

The DC gain for quantization noise is given by

$$\left.\frac{\mathrm{E}_{\mathrm{Q,Total}}(z)}{\mathrm{E}_{\mathrm{PQ}}(z)}\right|_{z=1} = \frac{1}{\mathrm{K}_{\mathrm{PQ}}\mathrm{T}_{\mathrm{PQ}}} \tag{6.12}$$

At low frequencies, quantization noise power spectral density referred to $\Phi_{\mathrm{ERR}}(z)$ input, $\mathrm{E}_{\mathrm{Q,In}}(z)$, is given by

$$\mathrm{E}_{\mathrm{Q,In}}(z) = \frac{\mathrm{E}_{\mathrm{PQ}}(z)}{\mathrm{K}_{\mathrm{PQ}}} \quad \mathrm{sec}^2/\mathrm{Hz} \tag{6.13}$$

Clearly, a larger K_{PQ} results in lower input referred quantization noise. It is interesting to note that while the DC gain of the input to output transfer function does not depend on phase quantizer gain, both bandwidth of the transfer function and noise suppression do depend on phase quantizer gain. The effective linear gain of the 1-bit phase quantizer can be calculated using the following equation [111]:

$$\mathrm{K}_{\mathrm{PQ}} = \frac{E\{sgn(\Delta t)\Delta t\}}{E\{\Delta t^2\}} \tag{6.14}$$

where $\Delta t[n]$ is the phase error at the input of the phase quantizer. For small values of K_{FB}, the probability density function (pdf) of Δt is the same as that of the running sum of the second order digital $\Delta\Sigma$ modulator output. Figure 6.10 shows the pdf of the running sum of the second order digital $\Delta\Sigma$ output as well as pdf of phase quantizer input for $\mathrm{K}_{\mathrm{FB}} = 2^{-6}$. Using the pdf of the running sum of the digital $\Delta\Sigma$ output, we can calculate the effective

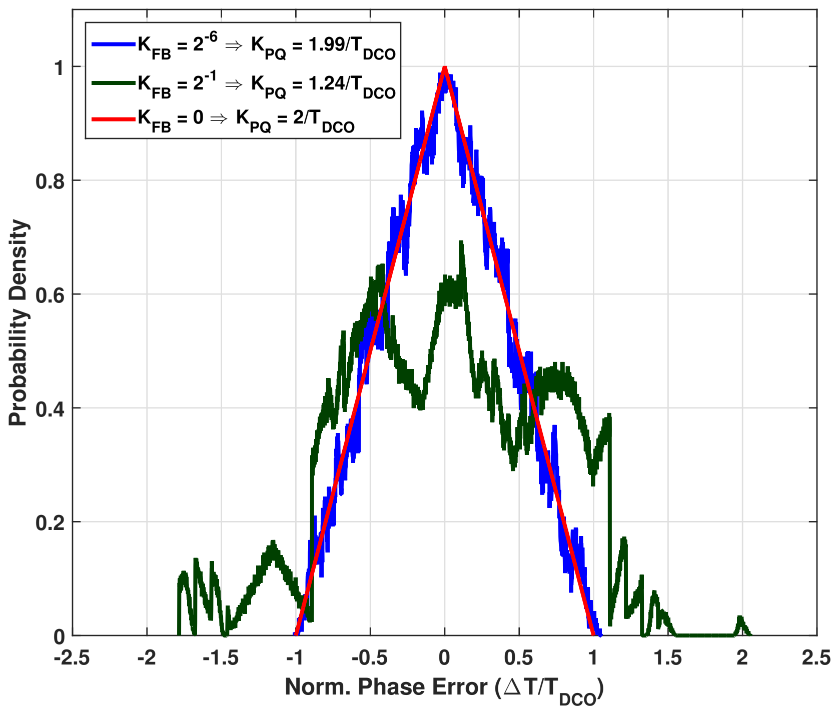


Figure 6.10: Probability density function of simulated phase quantizer input phase error for $\Delta\Sigma$ FDC with and without feedback.

linear gain of the phase quantizer to be

$$\mathrm{K_{PQ}} = \frac{2}{\mathrm{T_{DCO}}} \tag{6.15}$$

Large feedback gain $\mathrm{K_{FB}}$ results in larger input span for the phase quantizer as seen in Fig. 6.10. Therefore, to maximize $\mathrm{K_{PQ}}$, we should minimize $\mathrm{K_{FB}}$. There are two issues with reducing $\mathrm{K_{FB}}$ below a certain limit. First, the full-scale range of the $\Delta\Sigma$ FDC is $2\mathrm{T_{PQ}} = 2\mathrm{K_{FB}}\mathrm{T_{DCO}}$. A smaller value of $\mathrm{K_{FB}}$ reduces this full-scale range, which may result in increased in-band phase noise due to overload. Second, the STF bandwidth depends on $\mathrm{K_{FB}}$. The first order $\Delta\Sigma$ FDC can also be thought of as an integrator connected in a feedback loop. The continuous time approximation of the transfer function of this integrator is

$$\mathrm{H_I}(s) = \frac{\mathrm{K_{PQ}}}{s\mathrm{T_{REF}}} \tag{6.16}$$

When this integrator is connected in negative feedback configuration with feedback gain of $\mathrm{T_{PQ}}$, the DC gain of the resulting configuration is $1/\mathrm{T_{PQ}}$. The -3dB bandwidth of such

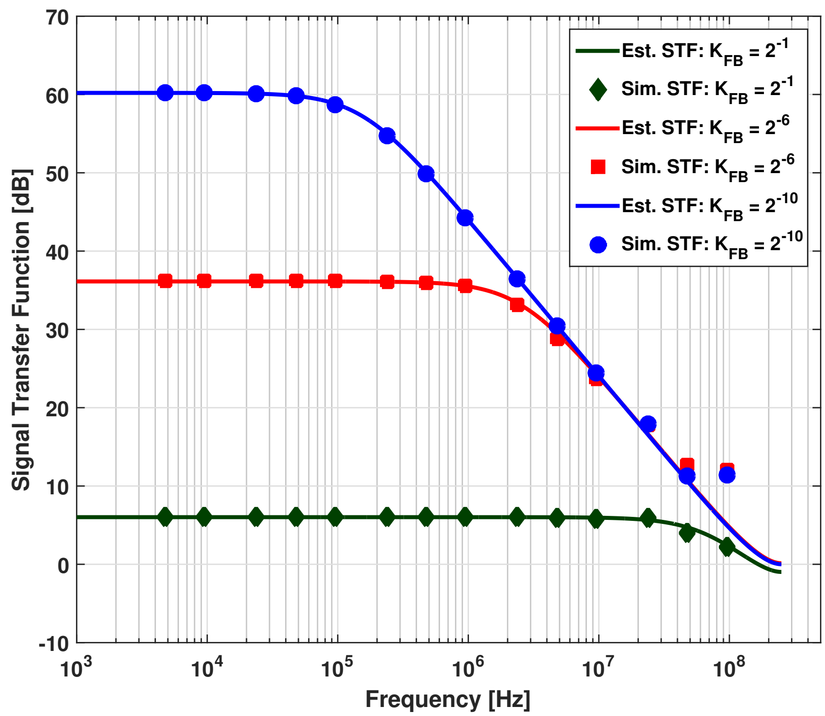


Figure 6.11: Simulated and estimated FDC input to output signal transfer function (STF) for various values of $\mathrm{K_{FB}}$.

configuration is given by

$$\omega_{\text{-3dB}} = \frac{\mathrm{K_{PQ}T_{PQ}}}{\mathrm{T_{REF}}} = \frac{\mathrm{K_{PQ}K_{FB}T_{DCO}}}{\mathrm{T_{REF}}} \tag{6.17}$$

Therefore, as $\mathrm{K_{FB}}$ reduces, bandwidth of the STF reduces. Note that any additional low frequency pole introduced by $\Delta\Sigma$ FDC STF may destabilize the overall FDCPLL loop. Therefore it is desirable to increase the bandwidth of $\Delta\Sigma$ FDC STF to achieve wide-band FDCPLL operation. Figure 6.11 shows the simulated and estimated input to output transfer functions for various values of $\mathrm{K_{FB}}$ which confirm our predictions. We observe that for low $\mathrm{K_{FB}}$, the phase quantizer gain mostly depends on digital $\Delta\Sigma$ quantization noise. When used in conjunction with an integer multi-modulus divider, the peak-to-peak value of the running sum of digital $\Delta\Sigma$ modulator is as large as $2\mathrm{T_{DCO}}$. This large input range of the phase quantizer limits the maximum value of $\mathrm{K_{PQ}}$ to $2/\mathrm{T_{DCO}}$.

To be able to use 1-bit first order $\Delta\Sigma$ FDC in a wide-band fractional-N PLL, a large $\mathrm{K_{PQ}}$, and $\mathrm{F_{REF}}$ are needed. Large $\mathrm{F_{REF}}$ also has an added benefit of reducing in-band

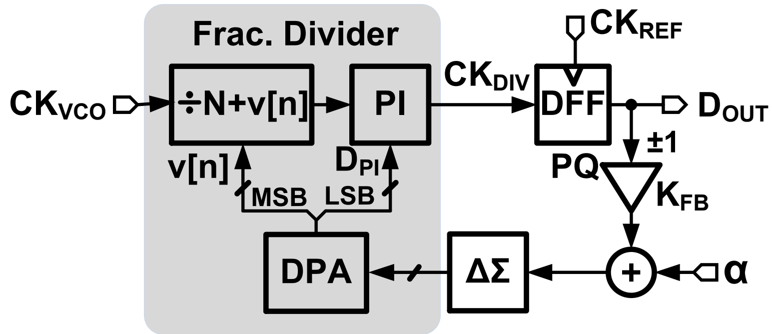


Figure 6.12: Simplified block diagram of proposed fractional divider based ΔΣ FDC.

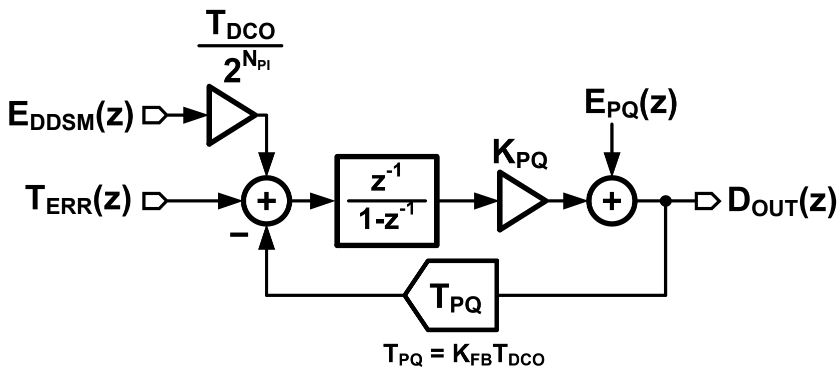


Figure 6.13: Simplified small signal equivalent block diagram for proposed PI-based ΔΣ FDC.

power spectral densities of both digital ΔΣ modulator quantization noise as well as PQ quantization noise. To increase K_{PQ}, it is necessary to reduce the input span of the phase quantizer. We propose a fractional divider based first order ΔΣ FDC that utilizes a PI for fractional division [112] resulting in phase quantizer input span reduction. We also present a two-stage PLL architecture that uses a first stage integer-N clock multiplier to increase the reference frequency input of second-stage fractional-N FDCPLL.

6.1.3 Proposed PI-based ΔΣ FDC

Simplified block diagram of the proposed ΔΣ FDC is shown in Fig. 6.12. The divider input is accumulated using a digital phase accumulator (DPA). The integer part of the DPA output is given to a MMD, while the fractional part is given to the PI. PI cancels the quantization error of the MMD and limits the input span of the phase quantizer to $T_{DCO}/2^{N_{PI}-1}$ for an N_{PI}

bit PI. Figure 6.13 shows a small signal equivalent block diagram of the proposed PI-based $\Delta\Sigma$ FDC. It is the same as that of gain-scaled FDC except for one crucial difference. The quantization error step for the digital $\Delta\Sigma$ modulator is reduced from $\mathrm{T_{DCO}}$ to $\mathrm{T_{DCO}}/2^{\mathrm{N_{PI}}}$. Note that the fractional divider contributes no additional quantization error if the digital $\Delta\Sigma$ modulator provides input with the same resolution as PI. Probability density function based on simulated histogram of the phase quantizer input phase error for a PI-based $\Delta\Sigma$ FDC is shown in Fig. 6.14. The input span of the phase quantizer reduces by a factor of 64 in case of a PI-based $\Delta\Sigma$ FDC with 6-b PI resolution. Therefore the effective linear gain of the phase quantizer, $\mathrm{K_{PQ}}$, increases to

$$\mathrm{K_{PQ}} = \frac{2^{\mathrm{N_{PI}}+1}}{\mathrm{T_{DCO}}} \tag{6.18}$$

in case of an $\mathrm{N_{PI}}$ bit PI-based FDC when $\mathrm{K_{FB}}$ is small. Simulations indicate that calculated $\mathrm{K_{PQ}}$ changes from 126 to 45 when $\mathrm{K_{FB}}$ is increased from 2^{-10} to 2^{-6}. The lower limit on $\mathrm{K_{FB}}$ is decided by the required capture range as well as $1/f^2$ noise of the VCO in the PLL. The plot of estimated and simulated input to output signal transfer function is shown in Fig. 6.15. The increase in bandwidth when PI-based fractional divider is used is evident from this plot.

6.2 Proposed Architecture

Figure 6.16 shows the block diagram of the FDC-based fractional PLL architecture. The output of the $\Delta\Sigma$ FDC, $\mathrm{F_{ERR}}$, is accumulated to find the phase error, Φ_{ERR}. A conventional proportional-integral type of digital loop filter generates a digital control word for the DCO. A $\Delta\Sigma$ modulator based digital-to-analog converter followed by a low pass filter is used to generate the control voltage VC for a LC-VCO. A type-II loop is chosen for implementation as it offers superior suppression of the VCO flicker noise. Figure 6.17 shows the simplified small signal equivalent block diagram of the proposed FDCPLL. The small signal model derived for cascade of $\Delta\Sigma$ FDC and digital accumulator in the previous section is used to simplify the loop analysis of the FDCPLL. The digital phase error signal $\mathrm{E}(z)$ is given as

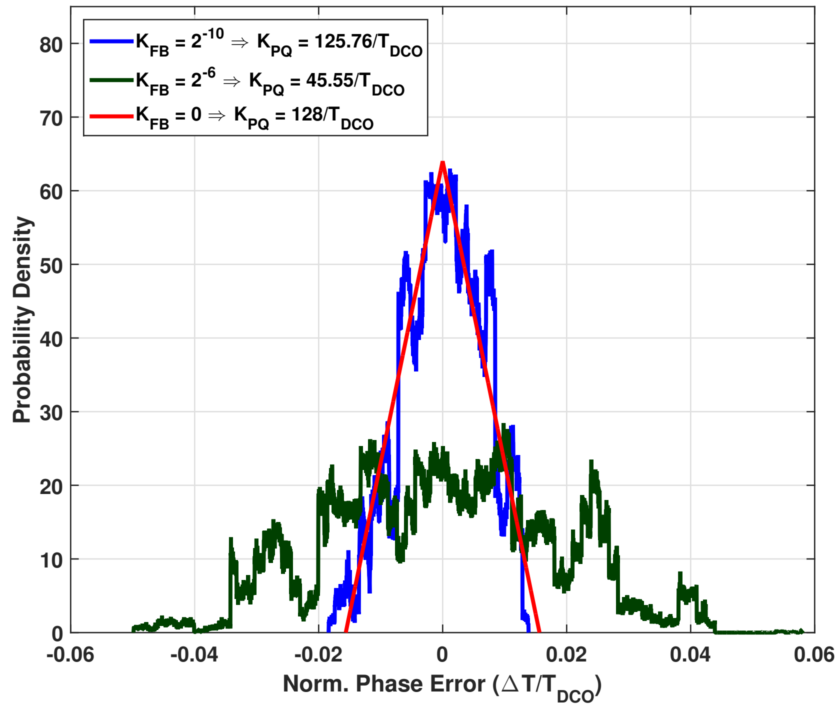


Figure 6.14: Probability density function of simulated phase quantizer input phase error for PI-based $\Delta\Sigma$ FDC.

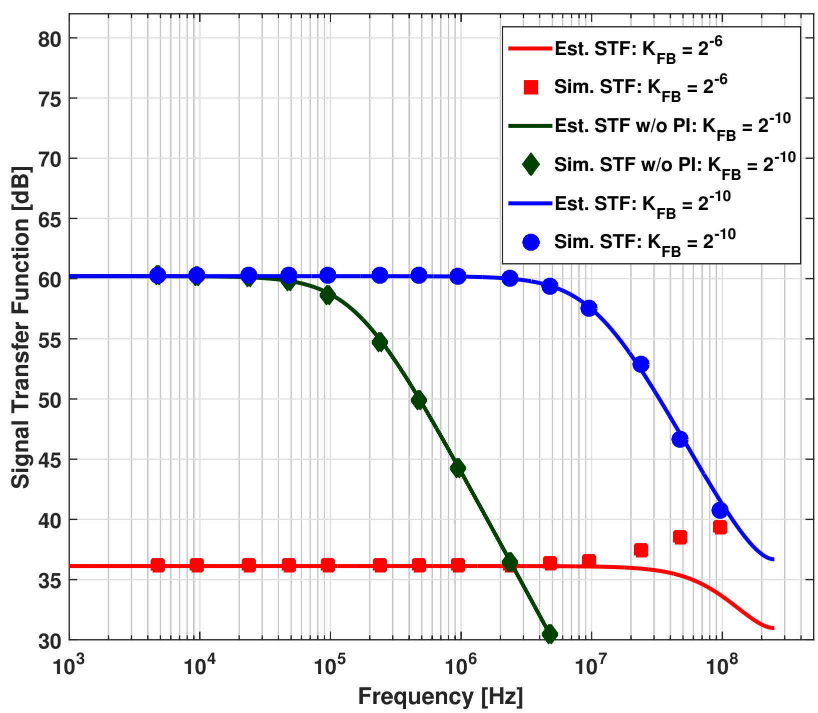


Figure 6.15: Simulated and estimated FDC input to output signal transfer function (STF) for various values of K_{FB} for a PI-based $\Delta\Sigma$ FDC.

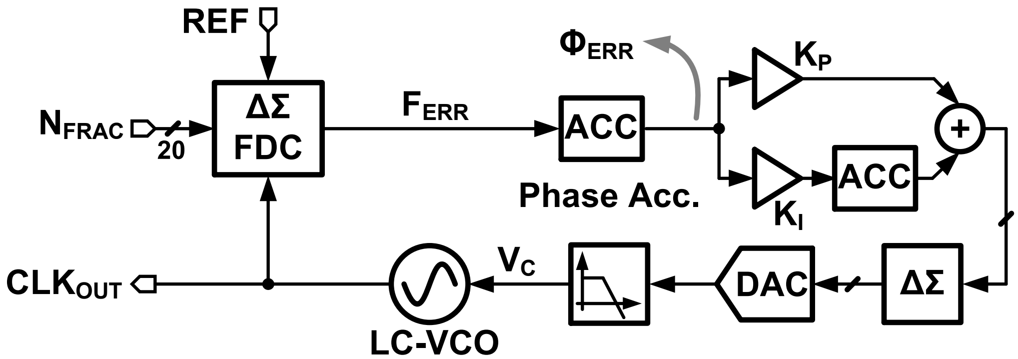


Figure 6.16: Block diagram of the proposed FDCPLL.

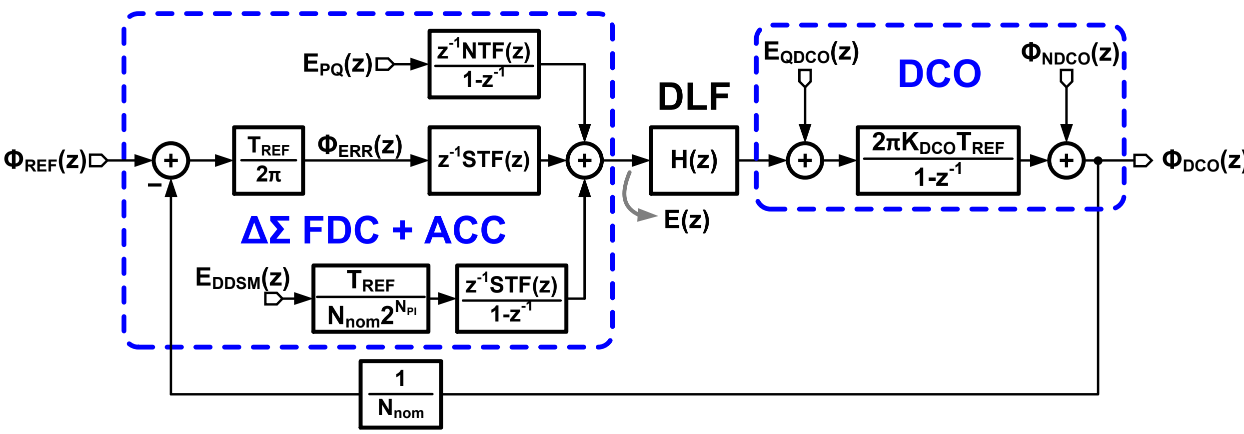


Figure 6.17: Simplified small signal equivalent block diagram of the proposed FDCPLL.

input to the digital loop filter with transfer function $\mathrm{H}(z)$. The output of DLF controls the DCO frequency. The frequency quantization error of the DCO is denoted as $\mathrm{E}_{\mathrm{QDCO}}(z)$ while the DCO phase noise is denoted as $\Phi_{\mathrm{NDCO}}(z)$. The DCO gain in units of Hz/LSB is denoted as $\mathrm{K}_{\mathrm{DCO}}$. The output of the FDCPLL is denoted as $\Phi_{\mathrm{DCO}}(z)$ and the reference input phase is denoted as $\Phi_{\mathrm{REF}}(z)$. $\mathrm{STF}(z)$ and $\mathrm{NTF}(z)$ are the signal transfer function and noise transfer function of the $\Delta\Sigma$ FDC, respectively. The loop gain of the FDCPLL is given by

$$\mathrm{LG}(z) = \frac{\mathrm{T}_{\mathrm{REF}}^2 \mathrm{K}_{\mathrm{DCO}}}{\mathrm{N}_{\mathrm{nom}}} \cdot \frac{z^{-1}}{1 - z^{-1}} \cdot \mathrm{STF}(z) \cdot \mathrm{H}(z) \tag{6.19}$$

Following the parameterization method described in [113], we define

$$\mathrm{G}(z) = \frac{\mathrm{LG}(z)}{1 + \mathrm{LG}(z)} \tag{6.20}$$

Let the power spectral densities of digital $\Delta\Sigma$ quantization noise, $\Delta\Sigma$ FDC quantization noise, DCO quantization noise, DCO phase noise, and reference phase noise be denoted as $\mathrm{S}_{\mathrm{Q,DDSM}}(z)$, $\mathrm{S}_{\mathrm{PQ}}(z)$, $\mathrm{S}_{\mathrm{Q,DCO}}(z)$, $\mathrm{S}_{\mathrm{N,DCO}}(z)$, and $\mathrm{S}_{\mathrm{N,REF}}(z)$ respectively. The overall output noise power spectral density is given by

$$\begin{aligned}
\mathrm{S}_{\mathrm{N,DCO}}(z) \;=\; & \left|\frac{2\pi}{2^{\mathrm{N}_{\mathrm{PI}}}} \cdot \mathrm{G}(z)\right|^2 \mathrm{S}_{\mathrm{Q,DDSM}}(z) + \left|\frac{2\pi}{\mathrm{T}_{\mathrm{REF}}} \cdot \frac{\mathrm{NTF}(z)}{(1 - z^{-1})\mathrm{STF}(z)} \cdot \mathrm{N}_{\mathrm{nom}} \cdot \mathrm{G}(z)\right|^2 \mathrm{S}_{\mathrm{PQ}}(z) \\
& + \left|\frac{2\pi \mathrm{K}_{\mathrm{DCO}} \mathrm{T}_{\mathrm{REF}}}{1 - z^{-1}} \cdot (1 - \mathrm{G}(z))\right|^2 \mathrm{S}_{\mathrm{Q,DCO}}(z) + \left|(1 - \mathrm{G}(z))\right|^2 \mathrm{S}_{\mathrm{N,DCO}}(z) \\
& + \left|\mathrm{N}_{\mathrm{nom}} \cdot \mathrm{G}(z)\right|^2 \mathrm{S}_{\mathrm{N,REF}}(z)
\end{aligned} \tag{6.21}$$

The in-band phase noise of this PLL is limited by the $\Delta\Sigma$ FDC quantization error, $\mathrm{E}_{\mathrm{PQ}}(z)$. The FDC quantization error can be reduced by increasing the reference frequency of the $\Delta\Sigma$ FDC-based PLL. As explained previously, the signal transfer function bandwidth of the $\Delta\Sigma$ depends on reference frequency. Therefore a larger reference frequency input to FDCPLL results in both wide-band operation as well as lower quantization noise. Larger reference frequency also pushes the quantization noise of the digital $\Delta\Sigma$ modulator present in $\Delta\Sigma$ FDC to higher frequencies [114]. The effect of using a larger input reference frequency on $\Delta\Sigma$ FDC signal transfer function is shown in Fig. 6.18. Behavioral simulations confirm that a higher reference frequency improves STF bandwidth of the $\Delta\Sigma$ FDC. To take advantage of improved performance of the $\Delta\Sigma$ FDC-based PLL with higher input reference frequency, we propose the use of a two-stage PLL architecture. The block diagram of the two-stage PLL is shown in Fig. 6.19. The first stage digital multiplying delay-locked loop (MDLL) generates a 500 MHz high frequency reference ($\mathrm{REF_{HF}}$) from a 31.25 MHz external crystal oscillator. The choice of MDLL is influenced by its excellent low frequency phase noise performance due to reference injection mechanism [39]. $\mathrm{REF_{HF}}$ is used as reference input for the proposed PI-based $\Delta\Sigma$ FDC in the second stage FDC-based PLL (FDCPLL). The FDC output is decimated by a factor of 4 to obtain a 10-bit frequency error ($\mathrm{F_{ERR}}$), which is accumulated to generate the phase error word (Φ_{ERR}). Φ_{ERR} is subsequently fed to a proportional-integral digital loop filter to achieve Type-II PLL response. A second order digital $\Delta\Sigma$ modulator truncates the 14-bit loop filter output to 5-bits and drives a current-mode DAC. A second order passive RC low-pass filter suppresses the shaped DAC quantization error and generates control voltage, $\mathrm{V_C}$, to tune the LC-VCO frequency. While first stage MDLL improves the performance of FDCPLL by increasing the reference frequency, it also adds phase noise to the overall system. Therefore, careful design of the MDLL is necessary for achieving good overall performance.

In the next section, we describe the implementation of important circuits blocks in the proposed PLL.

6.3 Implementation

The proposed PI-based $\Delta\Sigma$ FDC is implemented as shown in Fig. 6.20. The phase interpolator is implemented using a shift register-based Multi-Phase Generator (MPG) followed by a current-mode logic-based phase mixer [115]. This architecture avoids the need for quadrature phases and relaxes timing constraints for the phase interpolator control circuitry. MPG generates clock phases used for phase mixing (Φ_0, Φ_1) as well as clock signals used for clocking digital-to-analog converter inside phase mixer ($\mathrm{CK_{DAC}}$) and for clocking synthesized digital logic ($\mathrm{CK_{DIG}}$). These phases are generated from a low frequency input clock provided by multi-modulus divider. A D flip-flop implemented using double-tail latch type sense ampli-

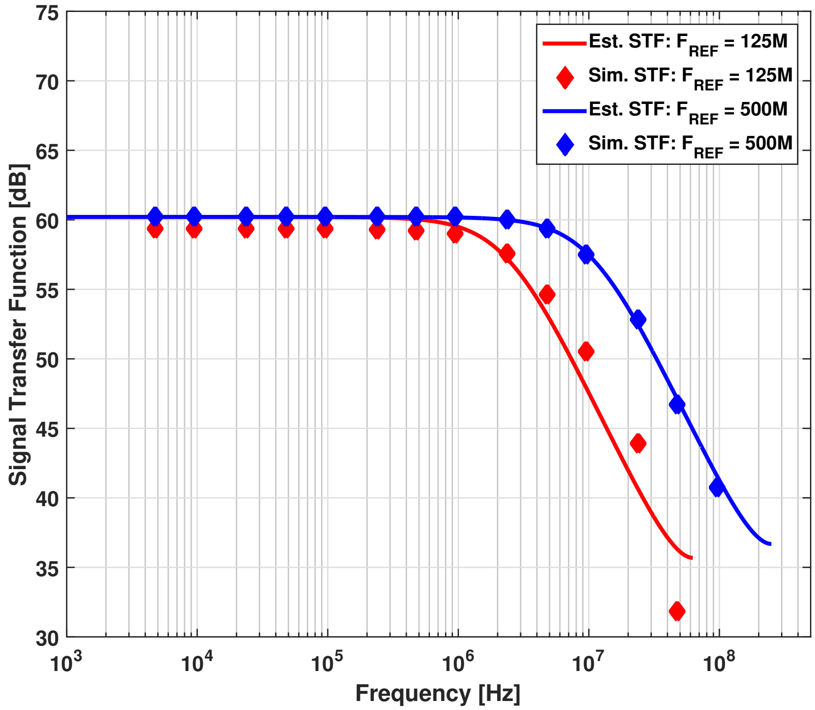


Figure 6.18: Simulated and estimated FDC input to output signal transfer function (STF) for various values of F_{REF} for a PI-based $\Delta\Sigma$ FDC.

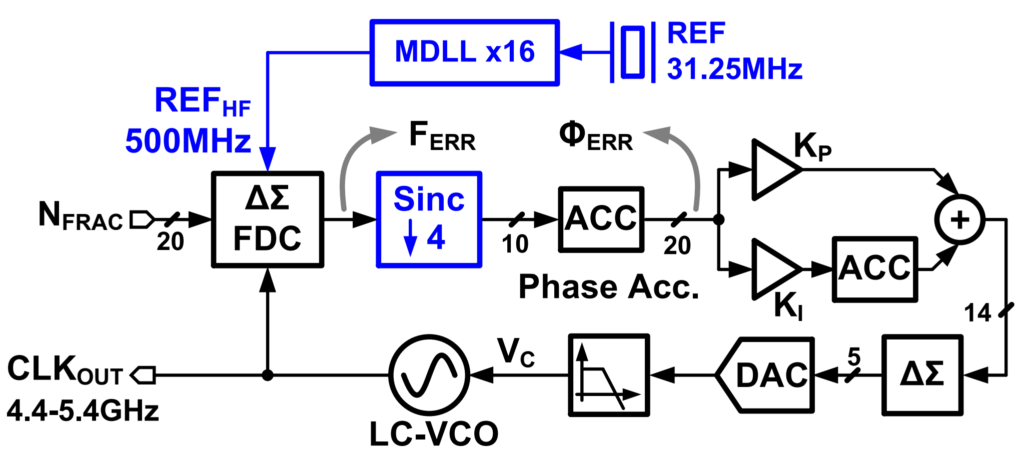


Figure 6.19: Block diagram of the proposed two-stage PLL.

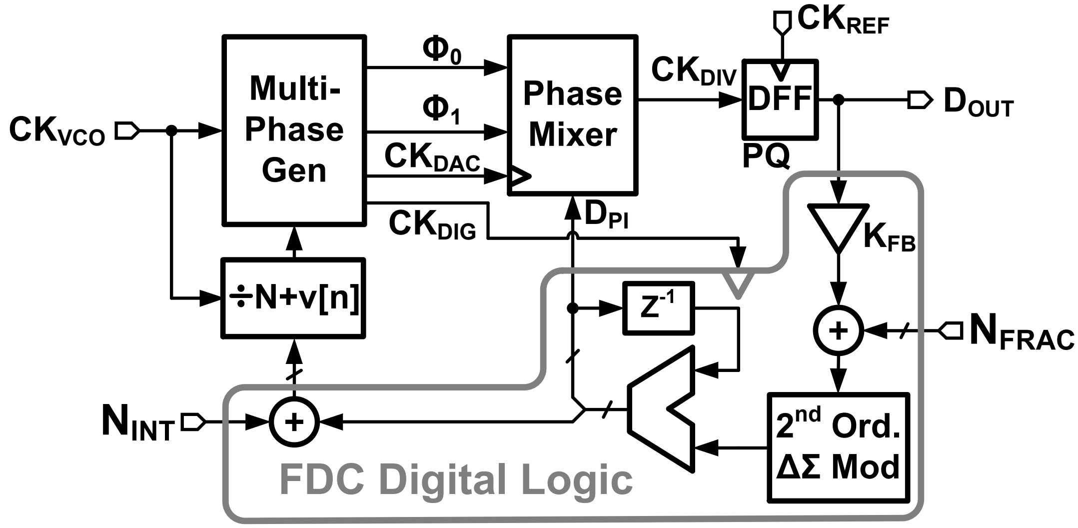


Figure 6.20: Block diagram of the proposed PI-based ΔΣ FDC implementation.

fier circuit [116] is used as 1-bit phase quantizer (PQ). The output of the phase quantizer is scaled and added to 20-bit fractional frequency control word N_{FRAC}. A second order digital ΔΣ modulator quantizes this input to generate a 6-bit output. An accumulator used as digital phase accumulator (DPA) generates control words for 6-bit phase mixer as well as multi-modulus integer divider. Integer division control word N_{INT} is added to DPA output before feeding it to the multi-modulus divider. It should be noted that this implementation introduces feedback loop delay of around 4 reference cycles. The effect of this loop delay is found to be negligible from behavioral simulations. Placing the phase mixer after the multi-modulus divider obviates the need for extra logic that would otherwise be needed for large phase shifts [99], albeit at the cost of worse linearity.

The simplified block diagram of the multi-phase generator is shown in Fig. 6.21. This shift register-based implementation generates coarse clock phases (Φ_0, Φ_1) that are subsequently used in the phase mixer for interpolation. In addition, clocks for synthesized digital logic (CK_{DIG}) and phase mixer DAC (CK_{DAC}) are also generated by the multi-phase generator. Output of the multi-modulus divider is sampled by using D flip-flops that are clocked by high frequency clock output from VCO. As a result the outputs of the shift register D flip-flops are ideally spaced by DCO period, T_{DCO}. In practice, the clock-to-Q delays of D flip-flops also influence the phase spacing. Maintaining the phase spacing between Φ_0 and Φ_1 is important

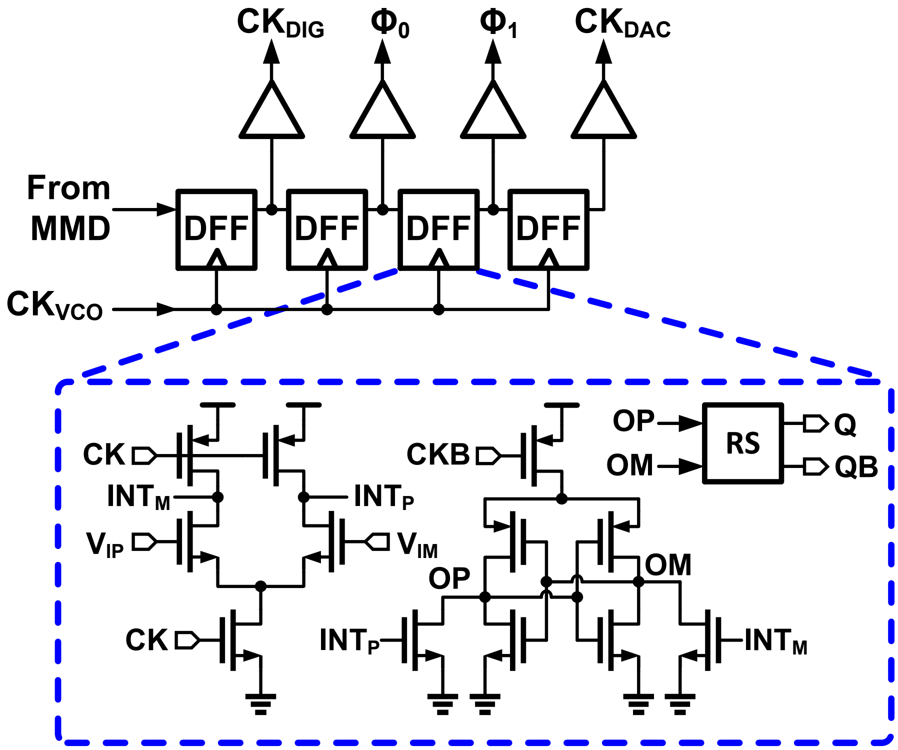


Figure 6.21: Block diagram of shift register-based multi-phase generator.

as these phases are used by the phase mixer for fine phase interpolation. Any spacing error between these phases contributes to gain error in the phase interpolator characteristic. To minimize the spacing error, clock-to-Q delay mismatch of these flip-flops is minimized by matching their input as well as output parasitic capacitance loading. Furthermore, the extra D flip-flops inserted at the beginning and at the end of the shift register ensure better matching between Φ_0 and Φ_1 waveforms. The outputs of these flip-flops are used for clocking the digital logic as well as phase mixer DAC. Note that the phase spacing is not very critical for these clock signals.

The phase mixer schematic is shown in Fig. 6.22. As described before, phases Φ_0 and Φ_1, which are T_{DCO} apart, are generated by the multi-phase generator. As the multi-modulus divider output waveform has a pulse width of $2T_{DCO}$, the waveforms for Φ_0 and Φ_1 inherit this pulse width. These pulses, spaced by T_{DCO}, are passed through slew rate control buffers to a CML-based phase mixer. The mixer performs phase interpolation between Φ_0 and Φ_1 according to the 6-bit control word D_{PI}. A 63-element thermometer-coded current steering DAC is used to control the interpolation weight in a monotonic fashion. Phase

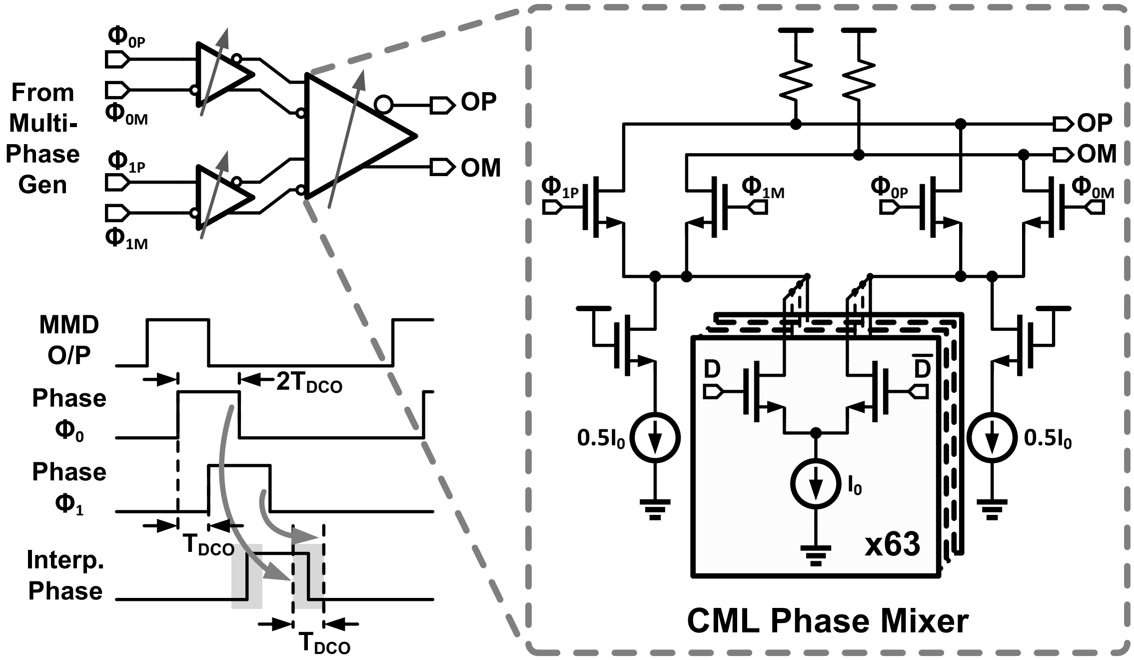


Figure 6.22: Block diagram of the phase mixer.

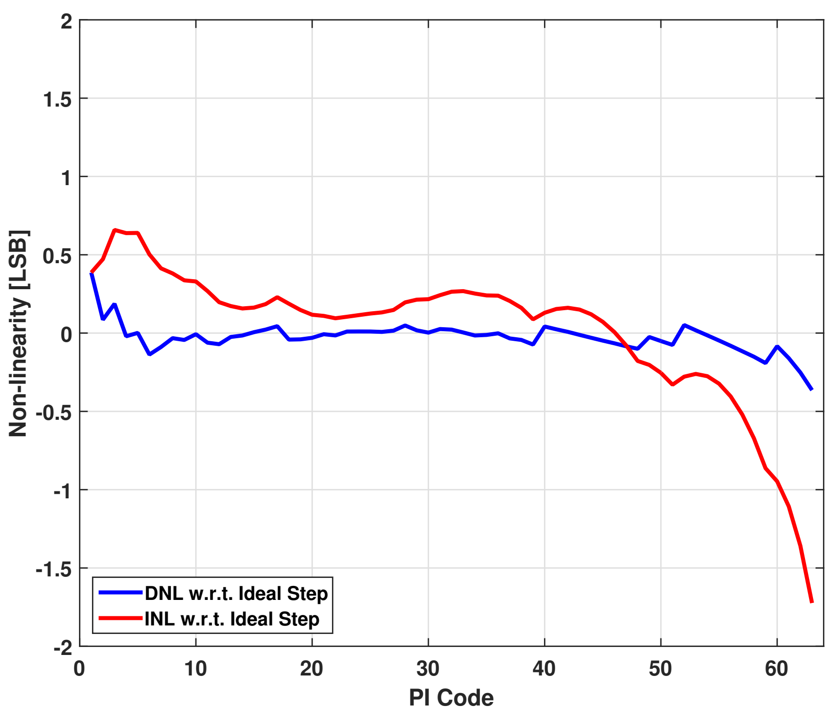


Figure 6.23: Simulated phase interpolator non-linearity.

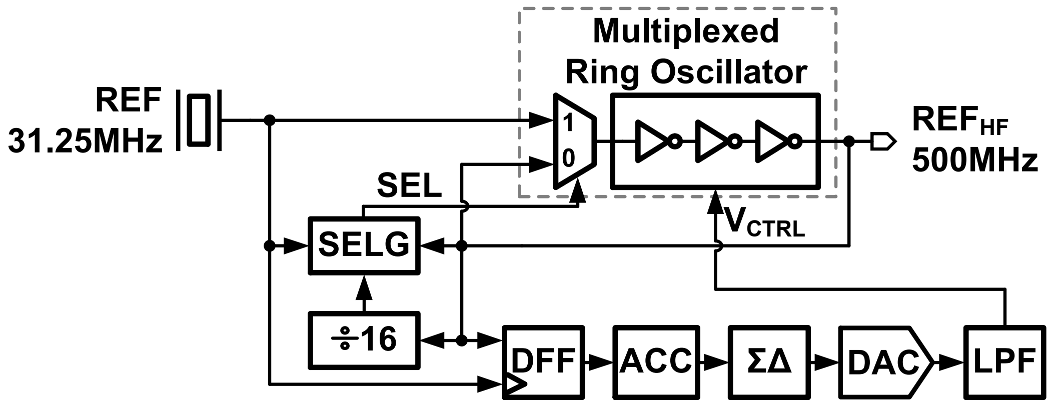


Figure 6.24: Block diagram of the digital MDLL.

interpolator linearity has a great impact on the spur performance of the FDCPLL. Therefore, interpolation linearity is improved by pre-distorting the DAC unit elements and controlling the rise/fall times of Φ_0 and Φ_1. Negative edges of Φ_0 and Φ_1 waveforms are used for interpolation as better linearity is achieved due to their vicinity to multiple transitions as opposed to positive edges. Two fixed half LSB current sources are used to improve phase mixer bandwidth. Figure 6.23 shows the simulated typical non-linear characteristic of the phase interpolator. Note that gain error is included in the plot as phase interpolator gain error also has adverse impact on FDCPLL performance.

The block diagram of the first stage digital MDLL is shown in Fig. 6.24 [117]. The MDLL utilizes a highly digital architecture. The reference edge is injected into the multiplexed ring oscillator using a select logic block, denoted as SELG and a divider. Low output frequency of the MDLL eases the design of the edge injection circuitry. A D flip-flop acts as a 1-bit phase quantizer to detect the phase difference between the reference input and the MDLL output. A digital accumulator followed by a $\Delta\Sigma$ modulator based DAC and low pass filter, LPF, generate the control voltage for the multiplexed ring oscillator. The edge injection mechanism of the MDLL achieves low jitter by suppressing the jitter accumulation in the multiplexed ring oscillator. Consequently low power operation is also achieved as less power needs to be burned in the ring oscillator.

For the FDCPLL, the digitally controlled oscillator is implemented using hybrid approach as described in [100]. The phase accumulator, decimation filter, and digital proportional-integral loop filter are implemented using automatic synthesis and place/route tools.

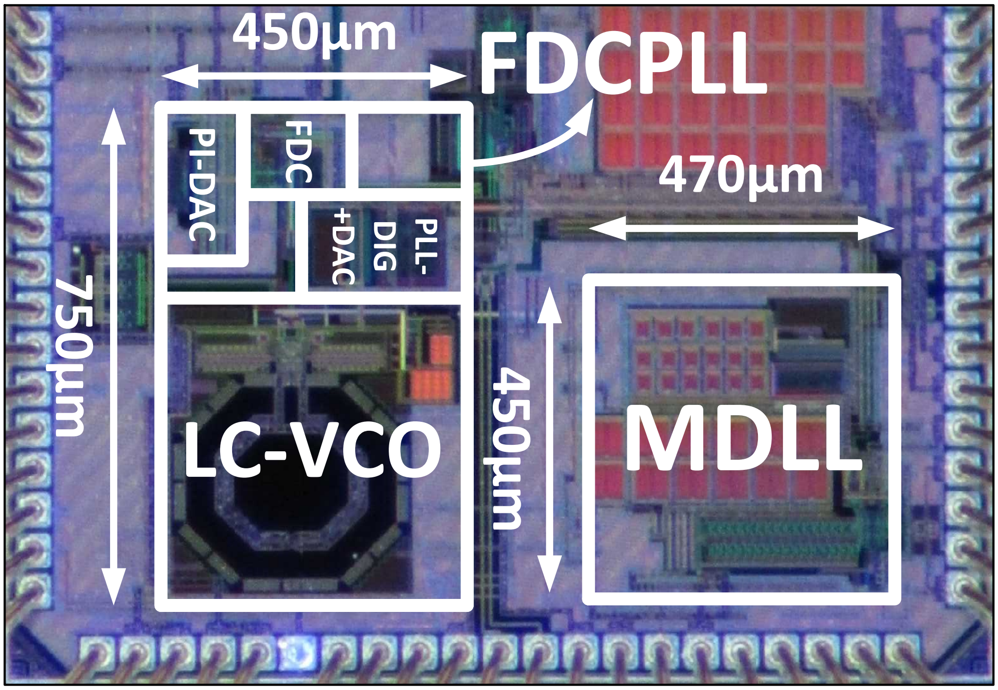


Figure 6.25: Die micrograph.

6.4 Measurement Results

The die micrograph of the proposed two-stage PLL is shown in Fig. 6.25. The prototype chip is fabricated in a 65 nm CMOS process. It operates on a supply voltage of 1V. The first stage MDLL and second stage FDCPLL occupy an active area of 0.22 mm^2 and 0.32 mm^2, respectively. The total power consumption when generating a 5.054 GHz output from a 31.25 MHz reference input is 10.1 mW, out of which the MDLL consumes 3 mW. The detailed breakdown of two-stage PLL power consumption is shown in Fig. 6.26.

We first describe the second stage FDCPLL measurements. To evaluate the performance of the 2nd stage $\Delta\Sigma$ FDC-based PLL, an external reference of 500 MHz is used. The impact of varying the $\Delta\Sigma$ feedback gain, K_{FB}, on FDCPLL performance at 5.25 GHz output frequency is shown in Fig. 6.27. At this frequency, the digital $\Delta\Sigma$ modulator input is zero in the present implementation. This is similar to measuring the performance of a TDC-based PLL at integer multiplication factors. From the measurement, we observe that for large K_{FB} values the FDCPLL loop phase margin is low resulting in peaking and limit cycles. This is attributed to reduced K_{PQ} and lower STF bandwidth for $\Delta\Sigma$ FDC. As K_{FB} is reduced,

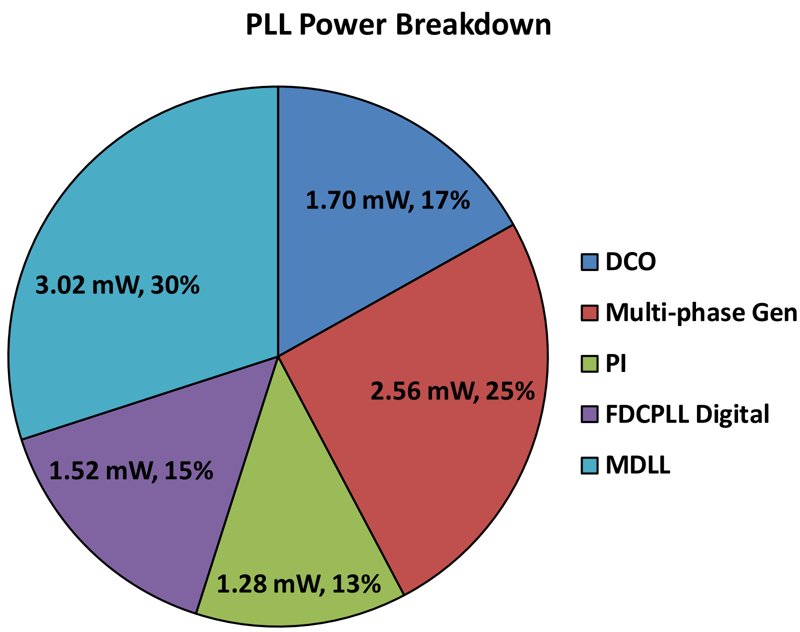


Figure 6.26: Detailed power breakdown of the two-stage PLL.

K_{PQ} as well as STF bandwidth increases. This results in improved phase margin as well as lower input referred quantization noise of the $\Delta\Sigma$ FDC. Consequently, the integrated jitter reduces from 3.69 ps_{rms} to 299 fs_{rms}, and in-band phase noise at 600 kHz offset reduces from -94.2 dBc/Hz to -109.3 dBc/Hz as K_{FB} is reduced from 2^{-3} to 2^{-10}. It is also interesting to note that for sufficiently low K_{FB}, FDCPLL bandwidth remains almost constant and independent of jitter as bandwidth of the $\Delta\Sigma$ FDC STF becomes large enough to achieve good phase margin for FDCPLL loop.

Figure 6.28 shows the PLL output phase noise spectra with different configurations at output frequency of 5.053955 GHz when the fractional spur is out-of-band. These configurations include a type I PLL with a gain-scaled $\Delta\Sigma$ FDC without PI, a type II PLL with a gain-scaled $\Delta\Sigma$ FDC without PI, and a type II PLL with the proposed PI-based $\Delta\Sigma$ FDC. The benefits of the type II loop are evident at low frequencies, as the PLLs using type II loop offer 41 dB higher suppression of the DCO flicker noise at 1 kHz. Furthermore, the large phase quantizer non-linearity present in type II PLL with gain-scaled $\Delta\Sigma$ FDC without PI, appears at the output in the form of increased in-band phase noise and peaking. For exactly the same loop parameters, the proposed PI-based $\Delta\Sigma$ FDC results in 14 dB better noise floor as well as wider bandwidth compared to gain-scaled $\Delta\Sigma$ FDC. The PLL with gain-scaled

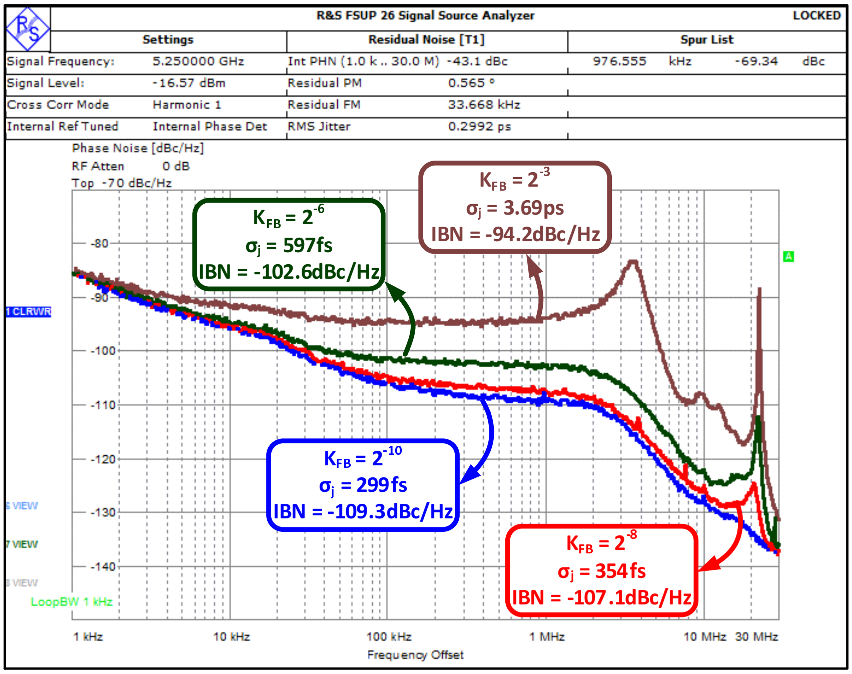


Figure 6.27: Measured output phase noise, integrated jitter (σ_j), and in-band phase noise floor (IBN) for various K_{FB} values when proposed FDCPLL operates at output frequency of 5.25 GHz.

$\Delta\Sigma$ FDC exhibits peaking as high as 26 dB compared to the noise floor obtained using the proposed PI-based $\Delta\Sigma$ FDC. The type I PLL shows an integrated jitter of 12.72 ps_{rms} in the frequency range of 1 kHz to 30 MHz due to inferior suppression of DCO noise. Type II PLL jitter using gain scaled FDC without PI is 1.67 ps_{rms} while the PLL jitter using the proposed PI-based FDC is 375 fs_{rms} in the frequency range of 1 kHz to 30 MHz. The PLL also achieves low in-band phase noise of -106.1 dBc/Hz using the proposed PI-based FDC. Figure 6.29 shows the PLL output phase noise spectra with different configurations at output frequency of 5.031738 GHz when the fractional spur is in-band. Again, the benefits of the type II loop are evident at low frequencies, as the PLLs using type II loop offer 39 dB higher suppression of the DCO flicker noise at 1 kHz. Furthermore, the large phase quantizer non-linearity present in type II PLL with gain-scaled $\Delta\Sigma$ FDC without PLL, appears at the output in the form of increased in-band noise floor and peaking. For exactly the same loop parameters, the proposed PI-based $\Delta\Sigma$ FDC results in 13 dB better noise floor as well as wider bandwidth compared to gain-scaled $\Delta\Sigma$ FDC. The PLL with gain-scaled $\Delta\Sigma$ FDC exhibits peaking as high as 25 dB compared to the noise floor obtained using the proposed PI-based $\Delta\Sigma$ FDC.

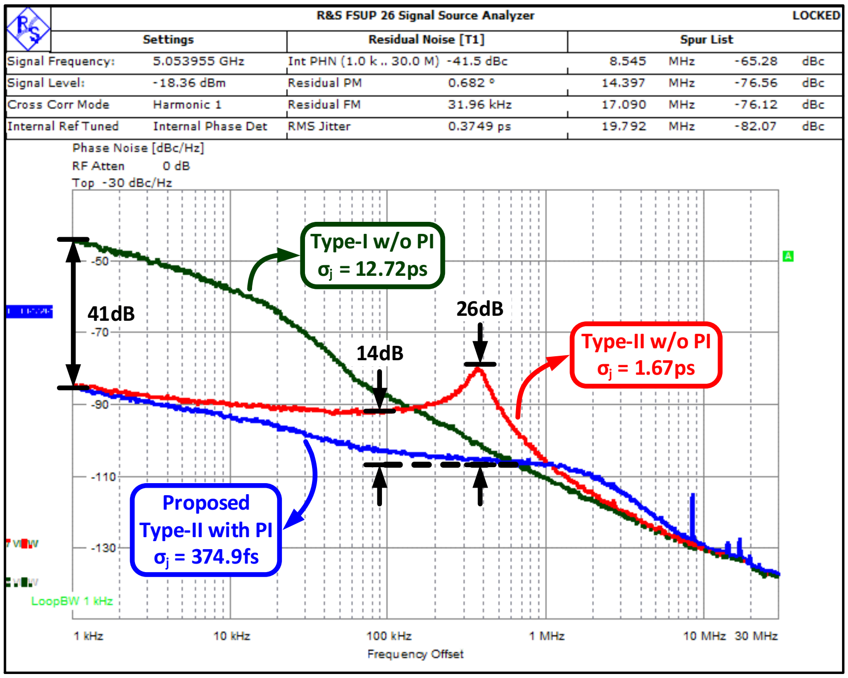


Figure 6.28: Measured output phase noise for various FDCPLL configurations for output frequency of 5.053955 GHz.

The type I PLL shows an integrated jitter of 10.1 ps_{rms} in the frequency range of 1 kHz to 30 MHz due to inferior suppression of VCO noise. Type II PLL jitter using gain scaled FDC without PI is 1.59 ps_{rms} while the PLL jitter using the proposed PI-based FDC is 404 fs_{rms} in the frequency range of 1 kHz to 30 MHz. The PLL also achieves low in-band phase noise of -106 dBc/Hz using the proposed PI-based FDC. The in-band fractional spur at 488 kHz has a strength of -51.4 dBc in the case of the proposed PI-based FDC, which increases to -44.1 dBc when PI is not used.

Fractional codes are swept starting from offset frequency of 4.875 GHz to measure the integrated jitter and spur performance of the FDCPLL as shown in Fig. 6.30 and Fig. 6.31, respectively. The integrated jitter varies between 464 fs_{rms} and 2.43 ps_{rms}, while the maximum spur strength varies between -31 dBc and -59 dBc. When the digital $\Delta\Sigma$ output is tone-free, we obtain the integrated jitter and spur performance as plotted in Fig. 6.32 and Fig. 6.33, respectively, for frequencies offset from 5.03125 GHz. In this case the integrated jitter varies between 373 fs_{rms} and 409 fs_{rms}, while the maximum spur strength varies between -53 dBc and -80 dBc.

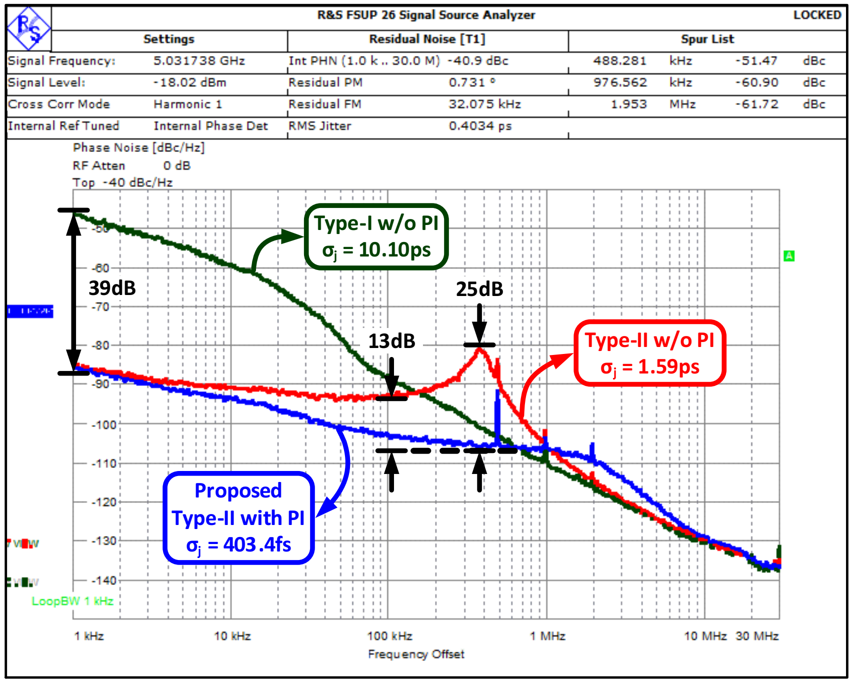


Figure 6.29: Measured output phase noise for various FDCPLL configurations for output frequency of 5.031738 GHz.

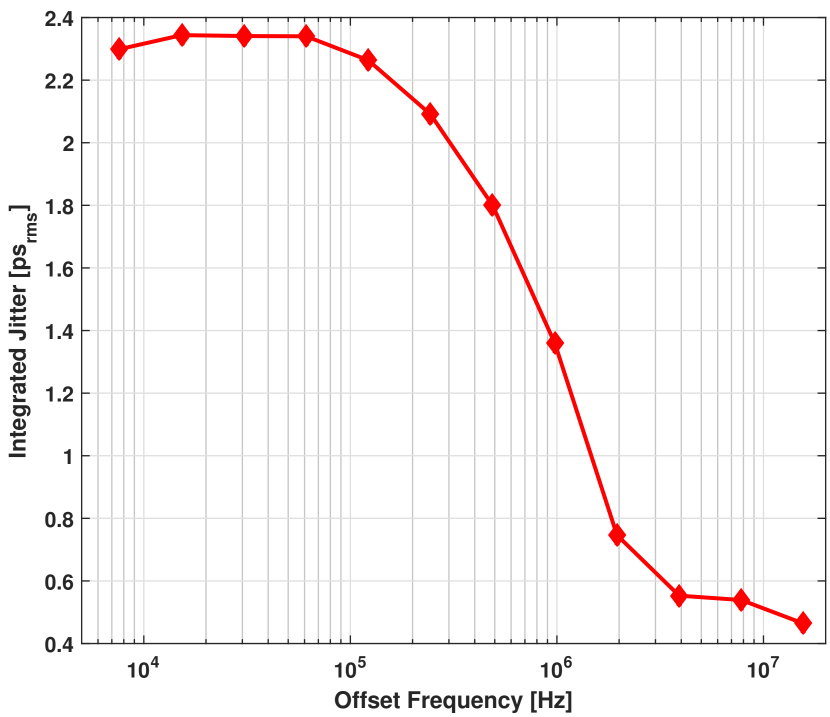


Figure 6.30: Measured output jitter integrated from 1 kHz to 30 MHz for fractional frequencies offset from 4.875 GHz.

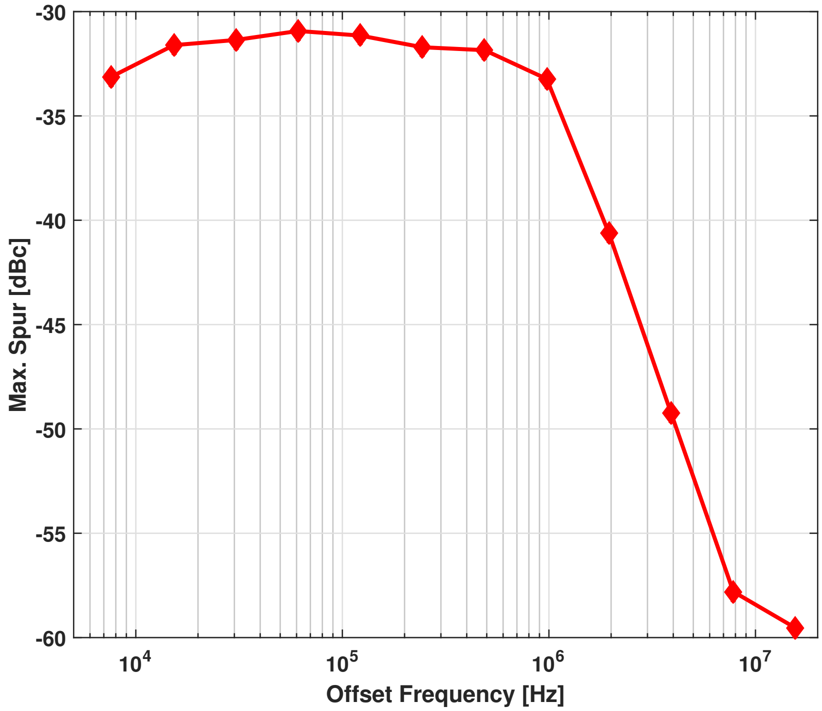


Figure 6.31: Measured maximum output spur for fractional frequencies offset from 4.875 GHz.

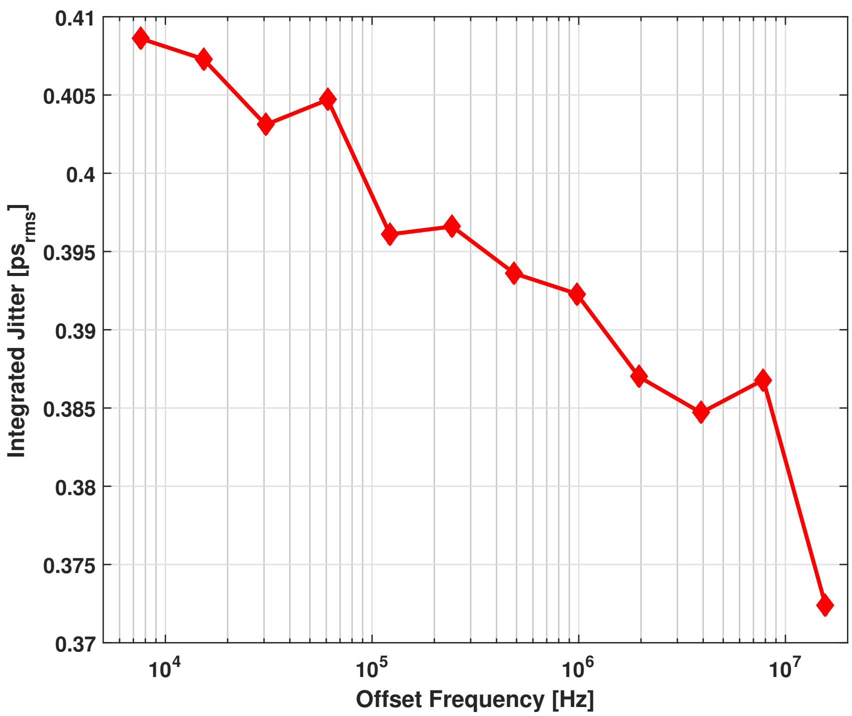


Figure 6.32: Measured output jitter integrated from 1 kHz to 30 MHz for fractional frequencies offset from 5.03125 GHz.

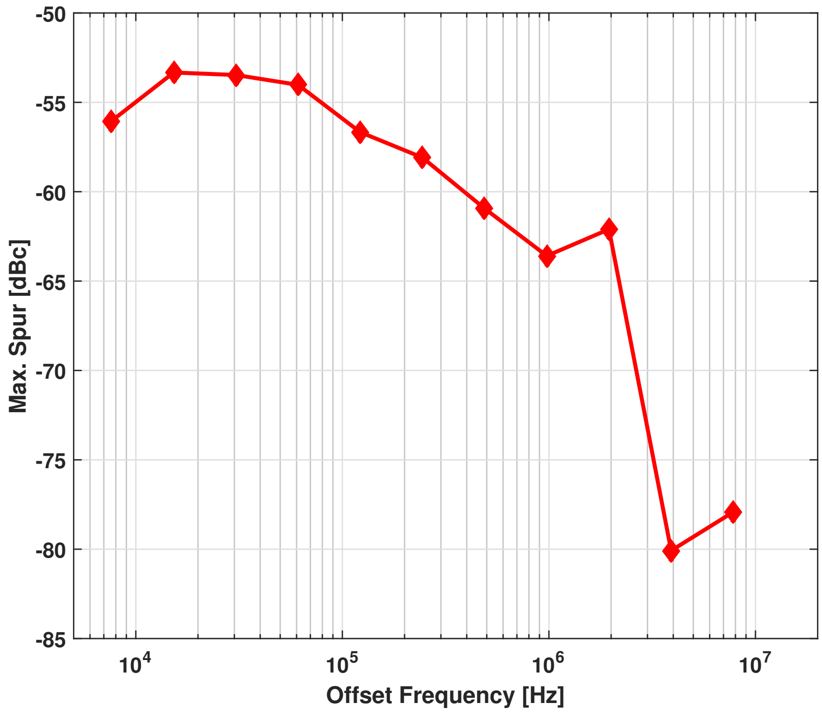


Figure 6.33: Measured maximum output spur for fractional frequencies offset from 5.03125 GHz.

The two-stage PLL achieves larger than 1 MHz bandwidth when generating 5.054 GHz output frequency from a 31.25 MHz crystal reference. The overall output phase noise of the proposed two-stage PLL is plotted in Fig. 6.34. The output phase noise of the 1st stage MDLL is also plotted. When integrated from 1 kHz to 30 MHz, the MDLL output jitter is 1.09 $\mathrm{ps_{rms}}$ whereas the overall two-stage PLL jitter is 848 $\mathrm{fs_{rms}}$. The increase in output jitter can be attributed to the increased low frequency phase noise from crystal reference as well as MDLL. The in-band phase noise level of -101.6 dBc/Hz at 600 kHz achieved by using a 1-bit $\Delta\Sigma$ FDC-based PLL corresponds to an equivalent TDC resolution of around 5 ps. It is also seen that MDLL output has a spur at 91 kHz that appears at the FDCPLL output with a strength of -59.3 dBc. This spur is believed to be caused by the $\Delta\Sigma$ DAC in the MDLL implementation. The overall output phase noise of the proposed two-stage PLL while generating output frequency of 5.031738 GHz is plotted in Fig. 6.35. This fractional frequency causes an in-band spur of -35 dBc at 488 kHz frequency. When integrated from 1 kHz to 30 MHz, the overall two-stage PLL jitter is 1.22 $\mathrm{ps_{rms}}$. The in-band phase noise level is still maintained at -101.6 dBc/Hz. This measurement indicates that the second stage

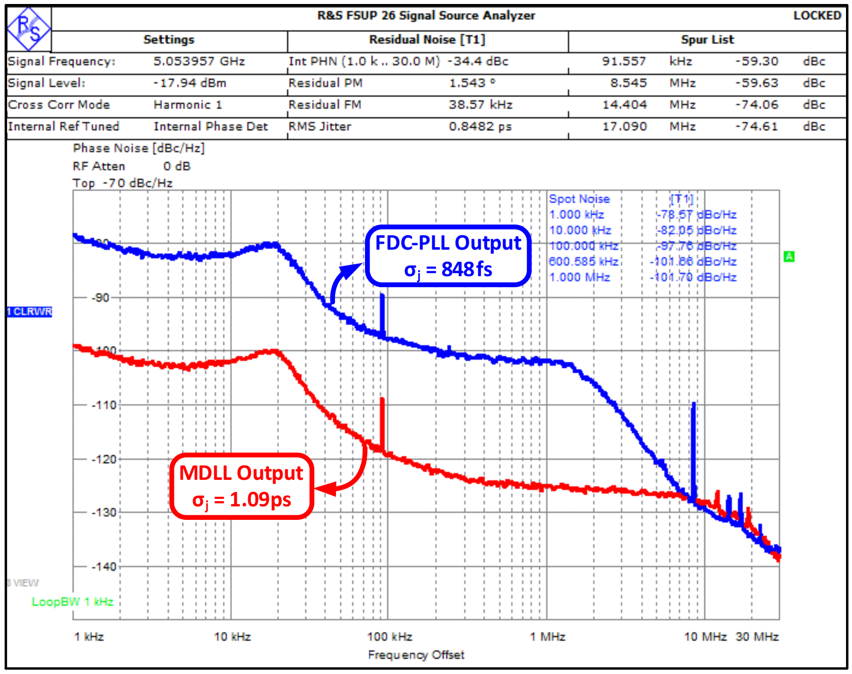


Figure 6.34: Measured two-stage PLL output phase noise for output frequency of 5.053955 GHz.

FDCPLL performance is sensitive to its reference input jitter.

Figure 6.36 and Fig. 6.37 show the output voltage spectrum of the MDLL and the second stage output, respectively, when the final output frequency is 5.053955 GHz. Reference spur of 48 dBc is measured at the MDLL output. The second stage FDCPLL further suppresses this reference spur to 77 dBc.

The performance of the proposed two-stage PLL and its comparison with other calibration-free digital fractional-N PLLs is shown in Table 6.1. The proposed PLL achieves more than 10 dB better normalized in-band noise floor as well as superior figure of merit compared to other calibration-free digital PLLs.

6.5 Conclusion

In summary, we presented a 1st order 1-bit $\Delta\Sigma$ FDC architecture that utilizes a PI-based fractional divider to mitigate phase quantizer non-linearities. As demonstrated by measurement results, this achieves in lower in-band phase noise and a wide bandwidth operation for

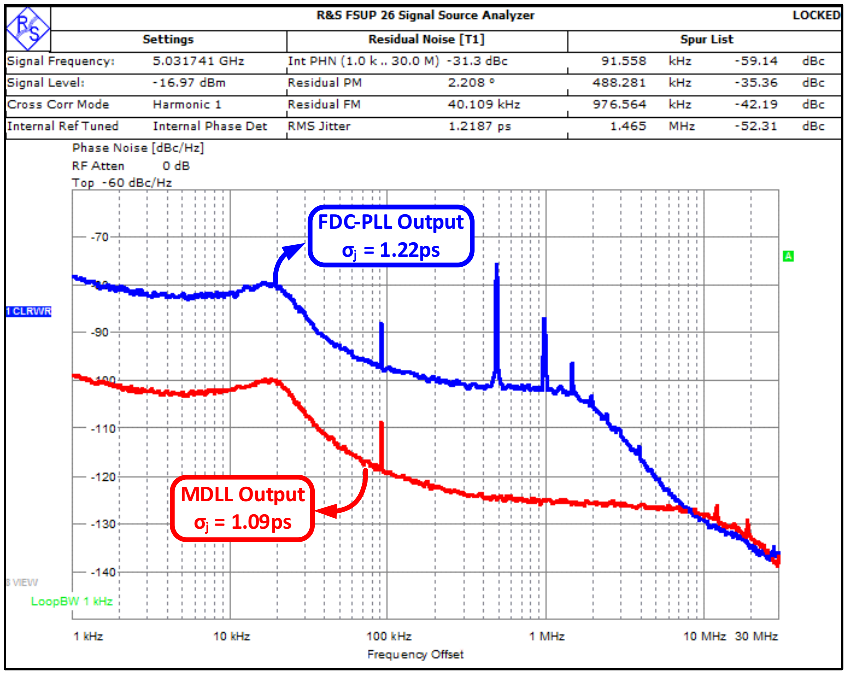


Figure 6.35: Measured two-stage PLL output phase noise for output frequency of 5.031738 GHz.

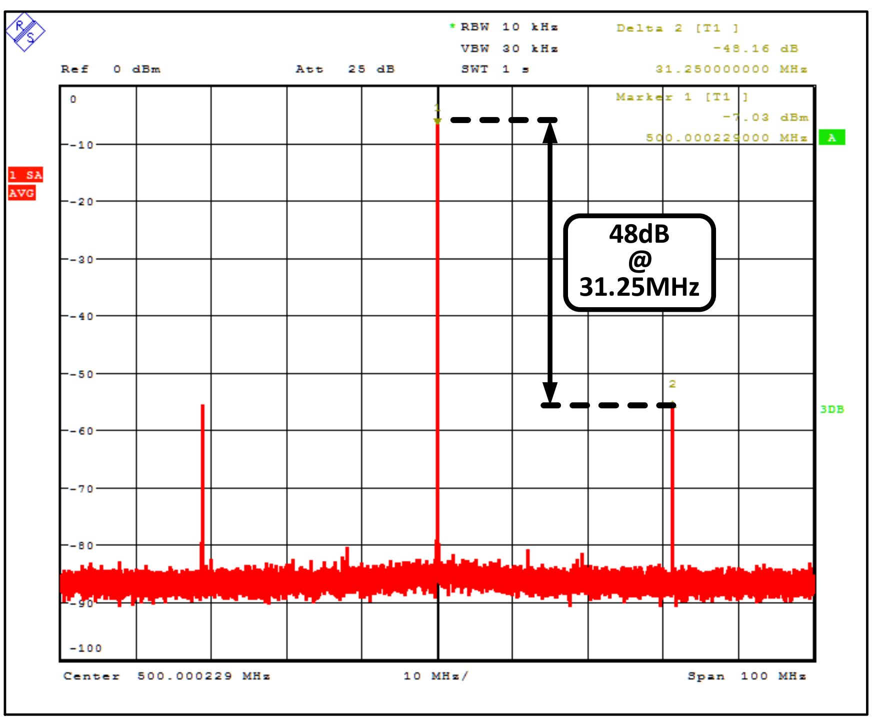


Figure 6.36: Measured MDLL output voltage spectrum.

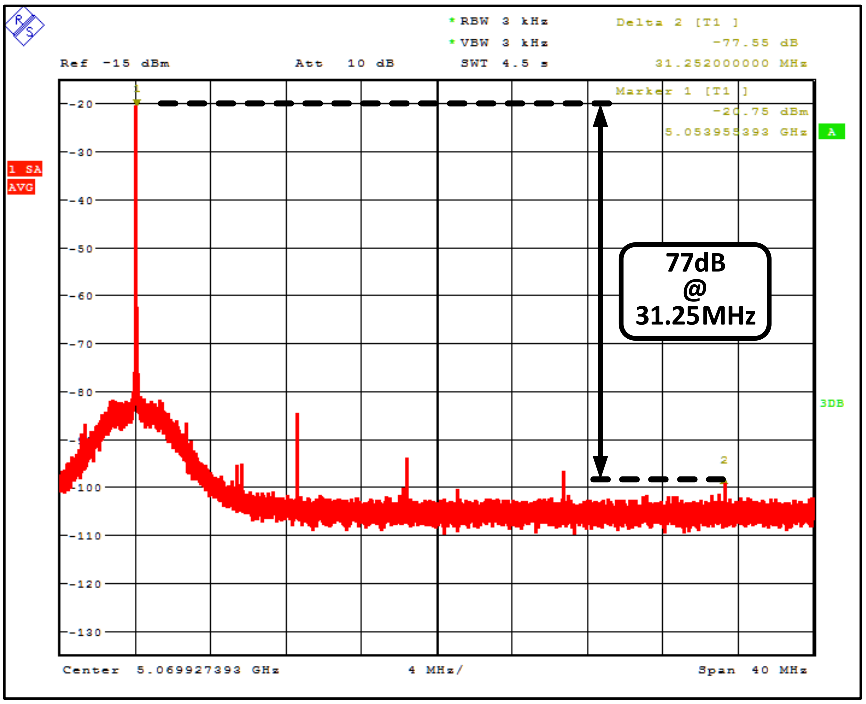


Figure 6.37: Measured output voltage spectrum of the proposed two-stage PLL.

Table 6.1: Performance comparison with calibration-free fractional-N PLLs.

	This Work		JSSC'08	ASSCC'12	JSSC'13
	2-Stage PLL	FDCPLL	[109]	[110]	[99]
Technology	65 nm		130 nm	65 nm	130 nm
Architecture	1-b ΔΣ FDC		1-b ΔΣ FDC	1-b ΔΣ FDC	MOBBPD
Supply [V]	1		1.4	1	1.3
Area [mm^2]	0.54	0.32	0.7	0.13	0.25
Ref. Freq. [MHz]	31.25	500	185.5	430	25
Output Freq. [GHz]	5.054		2.2	5.8	1.004
RMS Jitter [ps]	0.85	0.38	–	1.03	1.9
In-band Ph. Noise [dBc/Hz][a]	-101.6	-106.1	-77.8	-91.2	-91
Bandwidth [kHz]	1000		142	200	1000
Power [mW]	10.1	7.1	14	8	7.4
FoM [dB][b]	-231	-239.9	–	-230.7	-225.7

[a] Normalized to 5.054 GHz [b] FoM [dB] $= 10 \cdot \log_{10}((\sigma[\mathrm{s}])^2 \times (\mathrm{P[mW]}))$

a PLL. We also presented a two-stage calibration-free fractional-N PLL architecture that facilitates the use of $\Delta\Sigma$ FDC by increasing its input reference frequency.

CHAPTER 7

CONCLUSION

In this work, we analyzed and evaluated various system-level energy efficiency metrics for I/O interfaces. It is seen that rapid on-off I/O interfaces consume power in proportion with the data rate requirement without significantly degrading latency of the interface. It is important to achieve short turn-on and turn-off time for the interface circuitry to achieve this behavior.

Highly digital clock multipliers and use of digitally assisted analog circuits are proposed as the main techniques that allow interface circuitry to turn on and off rapidly. We demonstrated these techniques using a prototype rapid on-off transmitter chip. The proposed transmitter, owing to its short turn-on time, has power consumption proportional to its effective data rate, translating to an almost constant energy efficiency over a wide range of utilization levels. The transmitter operates over a 125X data rate range with power consumption that scales by 67X for 32-byte data bursts. A fast frequency settling DXRO architecture and a rapid on-off biasing circuit have been proposed to achieve this performance. The DXRO architecture uses resistor-based tuning and avoids the use of bias voltages for fast frequency settling. The fast charging technique used for ROOB circuit shows almost 30X improvement in its settling time compared to a diode-connected bias circuit. We have also proposed an analytical BER computation method to evaluate the impact of transmitter rapid on-off operation on the BER performance of the I/O interface using MDLL settling measurements and always-on transmitter measurements.

We have also proposed a phase presetting-based burst-mode receiver technique that utilizes link pulse response estimation to reduce the synchronization time. The highly digital nature and low analog hardware overhead of the proposed technique make it suitable for ADC-based links as well as links that utilize more complex modulation schemes such as 4-level pulse amplitude modulation (PAM4). A low complexity channel pulse response estimator using the LMS algorithm is also described. MATLAB as well as hardware implementation-based simulations were performed to verify the functionality of the estimator. This estimation technique can be utilized for implementing a phase presetting technique for burst-mode CDR for low power receivers operating over moderately lossy channels.

In the end, we presented a first order 1-bit $\Delta\Sigma$ FDC architecture that utilizes a PI-based fractional divider to mitigate phase quantizer non-linearities. As demonstrated by measurement results, the proposed $\Delta\Sigma$ FDC achieves lower in-band phase noise and a wide bandwidth for a PLL. We also presented a two-stage calibration-free fractional-N PLL architecture that facilitates the use of $\Delta\Sigma$ FDC by increasing its input reference frequency.

REFERENCES

[1] M. Mansuri, J. Jaussi, J. Kennedy, T.-C. Hsueh, S. Shekhar, G. Balamurugan, F. O'Mahony, C. Roberts, R. Mooney, and B. Casper, "A scalable 0.128-1 Tb/s, 0.8-2.6 pJ/bit, 64-lane parallel I/O in 32-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 48, no. 12, pp. 3229–3242, Dec 2013.

[2] W.-Y. Shin, G.-M. Hong, H. Lee, J.-D. Han, S. Kim, K.-S. Park, D.-H. Lim, J.-H. Chun, D.-K. Jeong, and S. Kim, "A 4.8 Gb/s impedance-matched bidirectional multi-drop transceiver for high-capacity memory interface," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2011, pp. 494–496.

[3] B. Casper, A. Martin, J. Jaussi, J. Kennedy, and R. Mooney, "An 8-Gb/s simultaneous bidirectional link with on-die waveform capture," *IEEE J. Solid-State Circuits*, vol. 38, no. 12, pp. 2111–2120, Dec. 2003.

[4] M. Bucher, R. Kollipara, B. Su, L. Gopalakrishnan, K. Prabhu, P. Venkatesan, K. Kaviani, B. Daly, B. Stonecypher, W. Dettloff, T. Stone, F. Heaton, Y. Lu, C. Madden, S. Bangalore, J. Eble, N. Nguyen, and L. Luo, "A 6.4-Gb/s near-ground single-ended transceiver for dual-rank DIMM memory interface systems," *IEEE J. Solid-State Circuits*, vol. 49, no. 1, pp. 127–139, Jan. 2014.

[5] HyperTransport Consortium. [Online]. Available: http://www.hypertransport.org/

[6] Peripheral Component Interconnect Special Interest Group. [Online]. Available: https://www.pcisig.com/specifications/pciexpress/

[7] B. Casper and F. O'Mahony, "Clocking analysis, implementation and measurement techniques for high-speed data links - a tutorial," *IEEE Trans. Circuits Syst. I*, vol. 56, no. 1, pp. 17–39, Jan. 2009.

[8] PCI Express 3.0 electrical. PCI-SIG. [Online]. Available: http://www.pcisig.com/developers/main/training_materials/

[9] M.-J. Lee, W. Dally, and P. Chiang, "Low-power area-efficient high-speed I/O circuit techniques," *IEEE J. Solid-State Circuits*, vol. 35, no. 11, pp. 1591–1599, Nov. 2000.

[10] J. Poulton, R. Palmer, A. Fuller, T. Greer, J. Eyles, W. Dally, and M. Horowitz, "A 14-mW 6.25-Gb/s transceiver in 90-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 42, no. 12, pp. 2745–2757, Dec. 2007.

[11] F. O'Mahony, J. E. Jaussi, J. Kennedy, G. Balamurugan, M. Mansuri, C. Roberts, S. Shekhar, R. Mooney, and B. Casper, "A 47×10 Gb/s 1.4 mW/Gb/s parallel interface in 45 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 45, no. 12, pp. 2828–2837, Dec. 2010.

[12] V. Krishnaswamy, D. Huang, S. Turullols, and J. Shin, "Bandwidth and power management of glueless 8-socket SPARC T5 system," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2013, pp. 58–59.

[13] R. Palmer, J. Poulton, A. Fuller, J. Chen, and J. Zerbe, "Design considerations for low-power high-performance mobile logic and memory interfaces," in *Proc. Asian Solid-State Circuits Conference A-SSCC*, Nov. 2008, pp. 205–208.

[14] L. Zhang, B. Ciftcioglu, M. Huang, and H. Wu, "Injection-Locked clocking: A new GHz clock distribution scheme," in *Proc. Custom Integrated Circuits Conference CICC*, Sep. 2006, pp. 785–788.

[15] K. Fukuda, H. Yamashita, G. Ono, R. Nemoto, E. Suzuki, N. Masuda, T. Takemoto, F. Yuki, and T. Saito, "A 12.3-mW 12.5-Gb/s complete transceiver in 65-nm CMOS process," *IEEE J. Solid-State Circuits*, vol. 45, no. 12, pp. 2838–2849, Dec. 2010.

[16] B. Casper, "Energy efficient multi-Gb/s I/O: Circuit and system design techniques," in *IEEE Workshop on Microelectronics and Electron Devices (WMED)*, April 2011.

[17] L. Barroso and U. Holzle, "The case for energy-proportional computing," *IEEE Computer*, vol. 40, no. 12, pp. 33–37, Dec. 2007.

[18] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding data center traffic characteristics," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 1, pp. 92–99, Jan. 2010.

[19] G.-Y. Wei, J. Kim, D. Liu, S. Sidiropoulos, and M. Horowitz, "A variable-frequency parallel I/O interface with adaptive power-supply regulation," *IEEE J. Solid-State Circuits*, vol. 35, no. 11, pp. 1600–1610, Nov. 2000.

[20] J. Kim and M. Horowitz, "Adaptive supply serial links with sub-1-V operation and per-pin clock recovery," *IEEE J. Solid-State Circuits*, vol. 37, no. 11, pp. 1403–1413, Nov. 2002.

[21] G. Balamurugan, J. Kennedy, G. Banerjee, J. Jaussi, M. Mansuri, F. O'Mahony, B. Casper, and R. Mooney, "A scalable 5-15 Gbps, 14-75 mW low-power I/O transceiver in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 43, no. 4, pp. 1010–1019, Apr. 2008.

[22] B. Leibowitz, R. Palmer, J. Poulton, Y. Frans, S. Li, J. Wilson, M. Bucher, A. Fuller, J. Eyles, M. Aleksic, T. Greer, and N. Nguyen, "A 4.3 GB/s mobile memory interface with power-efficient bandwidth scaling," *IEEE J. Solid-State Circuits*, vol. 45, no. 4, pp. 889–898, April 2010.

[23] J. Zerbe, B. Daly, W. Dettloff, T. Stone, W. Stonecypher, P. Venkatesan, K. Prabhu, B. Su, J. Ren, B. Tsang, B. Leibowitz, D. Dunwell, A. Carusone, and J. Eble, “A 5.6Gb/s 2.4mW/Gb/s bidirectional link with 8ns power-on,” in *Proc. Symposium on VLSI Circuits (VLSIC)*, June 2011, pp. 82–83.

[24] V. Soteriou and L.-S. Peh, “Exploring the design space of Self-Regulating Power-Aware On/Off interconnection networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 3, pp. 393–408, Mar. 2007.

[25] J. Luo, N. K. Jha, and L.-S. Peh, “Simultaneous dynamic voltage scaling of processors and communication links in Real-Time distributed embedded systems,” *IEEE Trans. VLSI Syst.*, vol. 15, no. 4, pp. 427–437, Apr. 2007.

[26] R. Marculescu, U. Y. Ogras, L.-S. Peh, N. E. Jerger, and Y. Hoskote, “Outstanding research problems in NoC design: System, microarchitecture, and circuit perspectives,” *IEEE Trans. Computer-Aided Design*, vol. 28, no. 1, pp. 3–21, Jan. 2009.

[27] T. Burd and R. Brodersen, “Design issues for dynamic voltage scaling,” in *Proc. Int. Symposium on Low Power Electronics and Design (ISLPED)*, July 2000, pp. 9–14.

[28] T. D. Burd, “Energy-efficient processor system design,” Ph.D. dissertation, University of California Berkeley, 2001.

[29] T. Sakurai and A. R. Newton, “Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas,” *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 584–594, Apr. 1990.

[30] A. Papoulis and S. Pillai, *Probability, Random Variables and Stochastic Processes*, 4th ed. Tata McGraw-Hill, 2002, ch. 16.

[31] M. Talegaonkar, A. Elshazly, K. Reddy, P. Prabha, T. Anand, and P. Hanumolu, “An 8 Gb/s - 64 Mb/s, 2.3 - 4.2 mW/Gb/s burst-mode transmitter in 90 nm CMOS,” *IEEE J. Solid-State Circuits*, vol. 49, no. 10, pp. 2228–2242, Oct. 2014.

[32] G.-Y. Tak, S.-B. Hyun, T. Y. Kang, B. G. Choi, and S. S. Park, “A 6.3-9-GHz CMOS fast settling PLL for MB-OFDM UWB applications,” *IEEE J. Solid-State Circuits*, vol. 40, no. 8, pp. 1671–1679, Aug. 2005.

[33] J. Dunning, G. Garcia, J. Lundberg, and E. Nuckolls, “An all-digital phase-locked loop with 50-cycle lock time suitable for high-performance microprocessors,” *IEEE J. Solid-State Circuits*, vol. 30, no. 4, pp. 412–422, April 1995.

[34] J. Lee and B. Kim, “A low-noise fast-lock phase-locked loop with adaptive bandwidth control,” *IEEE J. Solid-State Circuits*, vol. 35, no. 8, pp. 1137–1145, Aug. 2000.

[35] S. Dal Toso, A. Bevilacqua, M. Tiebout, S. Marsili, C. Sandner, A. Gerosa, and A. Neviani, “UWB fast-hopping frequency generation based on sub-harmonic injection locking,” *IEEE J. Solid-State Circuits*, vol. 43, no. 12, pp. 2844–2852, Dec. 2008.

[36] D. Dunwell, A. Carusone, J. Zerbe, B. Leibowitz, B. Daly, and J. Eble, "A 2.3-4 GHz injection-locked clock multiplier with 55.7% lock range and 10-ns power-on," in *Proc. IEEE Custom Integrated Circuits Conference (CICC)*, Sept. 2012, pp. 1–4.

[37] S. Drago, D. Leenaerts, B. Nauta, F. Sebastiano, K. Makinwa, and L. Breems, "A 200 μA duty-cycled PLL for wireless sensor nodes in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 45, no. 7, pp. 1305–1315, July 2010.

[38] R. Farjad-Rad, W. Dally, H.-T. Ng, R. Senthinathan, M.-J. Lee, R. Rathi, and J. Poulton, "A low-power multiplying DLL for low-jitter multigigahertz clock generation in highly integrated digital chips," *IEEE J. Solid-State Circuits*, vol. 37, no. 12, pp. 1804–1812, Dec. 2002.

[39] A. Elshazly, R. Inti, B. Young, and P. Hanumolu, "Clock multiplication techniques using digital multiplying delay-locked loops," *IEEE J. Solid-State Circuits*, vol. 48, no. 6, pp. 1416–1428, 2013.

[40] B. Helal, M. Straayer, G.-Y. Wei, and M. Perrott, "A highly digital MDLL-based clock multiplier that leverages a self-scrambling time-to-digital converter to achieve subpicosecond jitter performance," *IEEE J. Solid-State Circuits*, vol. 43, no. 4, pp. 855–863, April 2008.

[41] J. Tierno, A. Rylyakov, and D. Friedman, "A wide power supply range, wide tuning range, all static CMOS all digital PLL in 65 nm SOI," *IEEE J. Solid-State Circuits*, vol. 43, no. 1, pp. 42–51, Jan. 2008.

[42] D.-H. Oh, D.-S. Kim, S. Kim, D.-K. Jeong, and W. Kim, "A 2.8 Gb/s all-digital CDR with a 10b monotonic DCO," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2007, pp. 222–223.

[43] H. Song, D.-S. Kim, D.-H. Oh, S. Kim, and D.-K. Jeong, "A 1.0-4.0-Gb/s all-digital CDR with 1.0-ps period resolution DCO and adaptive proportional gain control," *IEEE J. Solid-State Circuits*, vol. 46, no. 2, pp. 424–434, Feb. 2011.

[44] J. Bulzacchelli, M. Meghelli, S. Rylov, W. Rhee, A. Rylyakov, H. Ainspan, B. Parker, M. Beakes, A. Chung, T. Beukema, P. Pepeljugoski, L. Shan, Y. Kwark, S. Gowda, and D. Friedman, "A 10-Gb/s 5-tap DFE/4-tap FFE transceiver in 90-nm CMOS technology," *IEEE J. Solid-State Circuits*, vol. 41, no. 12, pp. 2885–2900, Dec. 2006.

[45] C. Menolfi, T. Toifl, R. Reutemann, M. Ruegg, P. Buchmann, M. Kossel, T. Morf, and M. Schmatz, "A 25 Gb/s PAM4 transmitter in 90 nm CMOS SOI," in *IEEE ISSCC Dig. Tech. Papers*, vol. 1, Feb. 2005, pp. 72–73.

[46] F. J. Mesa-Martinez, E. K. Ardestani, and J. Renau, "Characterizing processor thermal behavior," in *Proc. Int. Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Mar. 2010, pp. 193–204.

[47] K. T. Malladi, I. Shaeffer, L. Gopalakrishnan, D. Lo, B. C. Lee, and M. Horowitz, "Rethinking DRAM power modes for energy proportionality," in *Proc. IEEE/ACM Int. Symposium on Microarchitecture (MICRO)*, Dec. 2012, pp. 131–142.

[48] T. Anand, M. Talegaonkar, A. Elshazly, B. Young, and P. Hanumolu, "A 2.5 GHz 2.2 mW/25 μW on/off-state power 2 ps_{rms}-long-term-jitter digital clock multiplier with 3-reference-cycles power-on time," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2013, pp. 256–257.

[49] T. Ali, A. Hafez, R. Drost, R. Ho, and C.-K. K. Yang, "A 4.6 GHz MDLL with -46 dBc reference spur and aperture position tuning," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2011, pp. 466–468.

[50] Analysis software for DSA8300 sampling oscilloscopes. Tektronix. [Online]. Available: http://www.tek.com/sites/tek.com/files/media/media/resources/61W_18868_6.pdf

[51] T. Beukema, M. Sorna, K. Selander, S. Zier, B. Ji, P. Murfet, J. Mason, W. Rhee, H. Ainspan, B. Parker, and M. Beakes, "A 6.4-Gb/s CMOS SerDes core with feed-forward and decision-feedback equalization," *IEEE J. Solid-State Circuits*, vol. 40, no. 12, pp. 2633–2645, Dec. 2005.

[52] A. Agrawal, J. Bulzacchelli, T. Dickson, Y. Liu, J. Tierno, and D. Friedman, "A 19-Gb/s serial link receiver with both 4-tap FFE and 5-tap DFE functions in 45-nm SOI CMOS," *IEEE J. Solid-State Circuits*, vol. 47, no. 12, pp. 3220–3231, Dec. 2012.

[53] J. Bulzacchelli, C. Menolfi, T. Beukema, D. Storaska, J. Hertle, D. Hanson, P.-H. Hsieh, S. Rylov, D. Furrer, D. Gardellini, A. Prati, T. Morf, V. Sharma, R. Kelkar, H. Ainspan, W. Kelly, L. Chieco, G. Ritter, J. Sorice, J. Garlett, R. Callan, M. Brandli, P. Buchmann, M. Kossel, T. Toifl, and D. Friedman, "A 28-Gb/s 4-tap FFE/15-tap DFE serial link transceiver in 32-nm SOI CMOS technology," *IEEE J. Solid-State Circuits*, vol. 47, no. 12, pp. 3232–3248, Dec. 2012.

[54] T. Toifl, C. Menolfi, M. Ruegg, R. Reutemann, D. Dreps, T. Beukema, A. Prati, D. Gardellini, M. Kossel, P. Buchmann, M. Brandli, P. Francese, and T. Morf, "A 2.6 mW/Gbps 12.5 Gbps RX with 8-tap switched-capacitor DFE in 32 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 47, no. 4, pp. 897–910, April 2012.

[55] P. Francese, T. Toifl, P. Buchmann, M. Brandli, C. Menolfi, M. Kossel, T. Morf, L. Kull, and T. Andersen, "A 16 Gb/s 3.7 mW/Gb/s 8-tap DFE receiver and baud-rate CDR with 31 kppm tracking bandwidth," *IEEE J. Solid-State Circuits*, vol. 49, no. 11, pp. 2490–2502, Nov. 2014.

[56] J. Han, Y. Lu, N. Sutardja, K. Jung, and E. Alon, "A 60Gb/s 173mW receiver frontend in 65nm CMOS technology," in *Proc. Symposium on VLSI Circuits (VLSIC)*, June 2015, pp. C230–C231.

[57] M. Harwood, N. Warke, R. Simpson, T. Leslie, A. Amerasekera, S. Batty, D. Colman, E. Carr, V. Gopinathan, S. Hubbins, P. Hunt, A. Joy, P. Khandelwal, B. Killips, T. Krause, S. Lytollis, A. Pickering, M. Saxton, D. Sebastio, G. Swanson, A. Szczepanek, T. Ward, J. Williams, R. Williams, and T. Willwerth, "A 12.5Gb/s SerDes in 65nm CMOS using a baud-rate ADC with digital receiver equalization and clock recovery," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2007, pp. 436–591.

[58] J. Cao, B. Zhang, U. Singh, D. Cui, A. Vasani, A. Garg, W. Zhang, N. Kocaman, D. Pi, B. Raghavan, H. Pan, I. Fujimori, and A. Momtaz, "A 500 mW ADC-based CMOS AFE with digital calibration for 10 Gb/s serial links over KR-backplane and multimode fiber," *IEEE J. Solid-State Circuits*, vol. 45, no. 6, pp. 1172–1185, June 2010.

[59] E.-H. Chen, R. Yousry, and C.-K. Yang, "Power optimized ADC-based serial link receiver," *IEEE J. Solid-State Circuits*, vol. 47, no. 4, pp. 938–951, April 2012.

[60] E. Tabasy, A. Shafik, K. Lee, S. Hoyos, and S. Palermo, "A 6 bit 10 GS/s TI-SAR ADC with low-overhead embedded FFE/DFE equalization for wireline receiver applications," *IEEE J. Solid-State Circuits*, vol. 49, no. 11, pp. 2560–2574, Nov. 2014.

[61] B. Zhang, A. Nazemi, A. Garg, N. Kocaman, M. Ahmadi, M. Khanpour, H. Zhang, J. Cao, and A. Momtaz, "A 40 nm CMOS 195 mW/55 mW dual-path receiver AFE for multi-standard 8.5-11.5 Gb/s serial links," *IEEE J. Solid-State Circuits*, vol. 50, no. 2, pp. 426–439, Feb. 2015.

[62] R. Navid, E.-H. Chen, M. Hossain, B. Leibowitz, J. Ren, C.-H. Chou, B. Daly, M. Aleksic, B. Su, S. Li, M. Shirasgaonkar, F. Heaton, J. Zerbe, and J. Eble, "A 40 Gb/s serial link transceiver in 28 nm CMOS technology," *IEEE J. Solid-State Circuits*, vol. 50, no. 4, pp. 814–827, April 2015.

[63] F. Spagna, L. Chen, M. Deshpande, Y. Fan, D. Gambetta, S. Gowder, S. Iyer, R. Kumar, P. Kwok, R. Krishnamurthy, C.-C. Lin, R. Mohanavelu, R. Nicholson, J. Ou, M. Pasquarella, K. Prasad, H. Rustam, L. Tong, A. Tran, J. Wu, and X. Zhang, "A 78mW 11.8Gb/s serial link transceiver with adaptive RX equalization and baud-rate CDR in 32nm CMOS," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2010, pp. 366 –367.

[64] K. Mueller and M. Muller, "Timing recovery in digital synchronous data receivers," *IEEE Trans. Commun.*, vol. 24, no. 5, pp. 516–531, May 1976.

[65] G. Gangasani, C.-M. Hsu, J. Bulzacchelli, T. Beukema, W. Kelly, H. Xu, D. Freitas, A. Prati, D. Gardellini, R. Reutemann, G. Cervelli, J. Hertle, M. Baecher, J. Garlett, P.-A. Francese, J. Ewen, D. Hanson, D. Storaska, and M. Meghelli, "A 32 Gb/s backplane transceiver with on-chip AC-coupling and low latency CDR in 32 nm SOI CMOS technology," *IEEE J. Solid-State Circuits*, vol. 49, no. 11, pp. 2474–2489, Nov. 2014.

[66] B. Leibowitz, J. Kizer, H. Lee, F. Chen, A. Ho, M. Jeeradit, A. Bansal, T. Greer, S. Li, R. Farjad-Rad, W. Stonecypher, Y. Frans, B. Daly, F. Heaton, B. Gariepp, C. Werner, N. Nguyen, V. Stojanovic, and J. Zerbe, “A 7.5Gb/s 10-tap DFE receiver with first tap partial response, spectrally gated adaptation, and 2nd-order data-filtered CDR,” in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2007, pp. 228–229.

[67] X.-Z. Qiu, X. Yin, J. Verbrugghe, B. Moeneclaey, A. Vyncke, C. Van Praet, G. Torfs, J. Bauwelinck, and J. Vandewege, “Fast synchronization 3R burst-mode receivers for passive optical networks,” *J. Lightwave Technol.*, vol. 32, no. 4, pp. 644–659, Feb. 2014.

[68] M. Banu and A. Dunlop, “A 660 Mb/s CMOS clock recovery circuit with instantaneous locking for NRZ data and burst-mode transmission,” in *IEEE ISSCC Dig.Tech. Papers*, Feb. 1993, pp. 102–103.

[69] J. Lee and M. Liu, “A 20-Gb/s burst-mode clock and data recovery circuit using injection-locking technique,” *IEEE J. Solid-State Circuits*, vol. 43, no. 3, pp. 619–630, Mar. 2008.

[70] J. Terada, K. Nishimura, S. Kimura, H. Katsurai, N. Yoshimoto, and Y. Ohtomo, “A 10.3 Gb/s burst-mode CDR using a $\Delta\Sigma$ DAC,” *IEEE J. Solid-State Circuits*, vol. 43, no. 12, pp. 2921–2928, Dec 2008.

[71] L. C. Cho, C. Lee, C. C. Hung, and S. I. Liu, “A 33.6-to-33.8 Gb/s burst-mode CDR in 90 nm CMOS technology,” *IEEE J. Solid-State Circuits*, vol. 44, no. 3, pp. 775–783, Mar. 2009.

[72] W.-S. Choi, T. Anand, G. Shu, A. Elshazly, and P. Hanumolu, “A burst-mode digital receiver with programmable input jitter filtering for energy proportional links,” *IEEE J. Solid-State Circuits*, vol. 50, no. 3, pp. 737–748, Mar. 2015.

[73] J. Kim and D.-K. Jeong, “Multi-gigabit-rate clock and data recovery based on blind oversampling,” *IEEE Commun. Mag.*, vol. 41, no. 12, pp. 68–74, Dec. 2003.

[74] H. Tagami, S. Kozaki, K. Nakura, S. Kohama, M. Nogami, and K. Motoshima, “A burst-mode bit-synchronization IC with large tolerance for pulse-width distortion for gigabit Ethernet PON,” *IEEE J. Solid-State Circuits*, vol. 41, no. 11, pp. 2555–2565, Nov. 2006.

[75] N. Suzuki, K. Nakura, S. Kozaki, H. Tagami, M. Nogami, and J. Nakagawa, “82.5 GSample/s (10.3125 GHz X 8 phase clocks) burst-mode CDR for 10G-EPON systems,” *Electronics Letters*, vol. 45, no. 24, pp. 1261–1263, Nov. 2009.

[76] T. Anand, M. Talegaonkar, A. Elkholy, S. Saxena, A. Elshazly, and P. Hanumolu, “A 7 Gb/s embedded clock transceiver for energy proportional links,” *IEEE J. Solid-State Circuits*, vol. 50, no. 12, pp. 3101–3119, Dec. 2015.

[77] A. Rylyakov, J. Proesel, S. Rylov, B. Lee, J. Bulzacchelli, A. Ardey, B. Parker, M. Beakes, C. Baks, C. Schow, and M. Meghelli, "A 25 Gb/s burst-mode receiver for low latency photonic switch networks," *IEEE J. Solid-State Circuits*, vol. 50, no. 12, pp. 3120–3132, Dec. 2015.

[78] K. Ohno and F. Adachi, "Fast clock synchroniser using initial phase presetting DPLL (IPP-DPLL) for burst signal reception," *Electronics Letters*, vol. 27, no. 21, pp. 1902–1904, Oct. 1991.

[79] M. Hossain, F. Aquil, P. S. Chau, B. Tsang, P. Le, J. Wei, T. Stone, B. Daly, C. Tran, J. Eble, K. Knorpp, and J. Zerbe, "A fast-lock, jitter filtering all-digital DLL based burst-mode memory interface," *IEEE J. Solid-State Circuits*, vol. 49, no. 4, pp. 1048–1062, Apr. 2014.

[80] Z. Xu, S. Lee, M. Miyahara, and A. Matsuzawa, "A 0.84ps-LSB 2.47mW time-to-digital converter using charge pump and SAR-ADC," in *Proc. IEEE Custom Integrated Circuits Conference (CICC)*, Sept. 2013, pp. 1–4.

[81] J. G. Proakis, *Digital Communications*, 4th ed. McGraw-Hill, 2001, ch. 10.

[82] M. Shanbhag. (2011, Apr.) 100 Gb/s simulated backplane channels. TE Connectivity. [Online]. Available: http://www.ieee802.org/3/100GCU/public/ChannelData/TEC_11_0428/TEC_STRADAWhisper29p8in_Meg6_Channel_IEEE802_3_100GbCu_04282011.zip

[83] Y. Hidaka, W. Gai, T. Horie, J. H. Jiang, Y. Koyanagi, and H. Osone, "A 4-channel 1.25 - 10.3 Gb/s backplane transceiver macro with 35 dB equalizer and sign-based zero-forcing adaptive control," *IEEE J. Solid-State Circuits*, vol. 44, no. 12, pp. 3547–3559, Dec. 2009.

[84] A. Momtaz and M. Green, "An 80 mW 40 Gb/s 7-tap T/2-spaced feed-forward equalizer in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 45, no. 3, pp. 629–639, Mar. 2010.

[85] S. Pavan, "A fixed transconductance bias technique for CMOS analog integrated circuits," in *Proc. Int. Symp. Circ. Syst.*, vol. 1, May 2004, pp. I–661–4 Vol.1.

[86] K.-L. Wong, A. Rylyakov, and C.-K. Yang, "A 5-mW 6-Gb/s quarter-rate sampling receiver with a 2-tap DFE using soft decisions," *IEEE J. Solid-State Circuits*, vol. 42, no. 4, pp. 881–888, Apr. 2007.

[87] H. Lee, C. H. Yue, S. Palermo, K. Mai, and M. Horowitz, "Burst mode packet receiver using a second order DLL," in *Proc. Symposium on VLSI Circuits (VLSIC)*, June 2004, pp. 264–267.

[88] V. Stojanovic, A. Ho, B. Garlepp, F. Chen, J. Wei, G. Tsang, E. Alon, R. Kollipara, C. Werner, J. Zerbe, and M. Horowitz, "Autonomous dual-mode (PAM2/4) serial link transceiver with adaptive equalization and data recovery," *IEEE J. Solid-State Circuits*, vol. 40, no. 4, pp. 1012–1026, Apr. 2005.

[89] S. Puligundla, F. Spagna, L. Chen, and A. Tran, "Validating the performance of a 32nm CMOS high speed serial link receiver with adaptive equalization and baud-rate clock data recovery," in *Proc. IEEE Int. Test Conference (ITC)*, Nov. 2010, pp. 1–5.

[90] A. Sanders, "Statistical simulation of physical transmission media," *IEEE Trans. Adv. Packag.*, vol. 32, no. 2, pp. 260 –267, May 2009.

[91] E.-H. Chen, J. Ren, B. Leibowitz, H.-C. Lee, Q. Lin, K. Oh, F. Lambrecht, V. Stojanovic, J. Zerbe, and C.-K. Yang, "Near-optimal equalizer and timing adaptation for I/O links using a BER-based metric," *IEEE J. Solid-State Circuits*, vol. 43, no. 9, pp. 2144–2156, Sept. 2008.

[92] T. Suttorp and U. Langmann, "A 10-Gb/s CMOS serial-link receiver using eye-opening monitoring for adaptive equalization and for clock and data recovery," in *Proc. IEEE Custom Integrated Circuits Conference (CICC)*, Sept. 2007, pp. 277–280.

[93] S. Gondi and B. Razavi, "Equalization and clock and data recovery techniques for 10-Gb/s CMOS serial-link receivers," *IEEE J. Solid-State Circuits*, vol. 42, no. 9, pp. 1999–2011, Sept. 2007.

[94] S. Son, H.-S. Kim, M.-J. Park, K. Kim, E.-H. Chen, B. Leibowitz, and J. Kim, "A 2.3-mW, 5-Gb/s low-power decision-feedback equalizer receiver front-end and its two-step, minimum bit-error-rate adaptation algorithm," *IEEE J. Solid-State Circuits*, vol. 48, no. 11, pp. 2693–2704, 2013.

[95] N. R. Shanbhag, *Signal Processing Kernels for Communications.* ECE 560 Class Notes, 2012, ch. 4.

[96] C.-M. Hsu, M. Straayer, and M. Perrott, "A low-noise wide-BW 3.6-GHz digital $\Delta\Sigma$ fractional-N frequency synthesizer with a noise-shaping time-to-digital converter and quantization noise cancellation," *IEEE J. Solid-State Circuits*, vol. 43, no. 12, pp. 2776–2786, Dec. 2008.

[97] C.-W. Yao and A. Wilson, "A 2.8-3.2-GHz fractional-N digital PLL with ADC-assisted TDC and inductively coupled fine-tuning DCO," *IEEE J. Solid-State Circuits*, vol. 48, no. 3, pp. 698–710, Mar. 2013.

[98] D. Tasca, M. Zanuso, G. Marzin, S. Levantino, C. Samori, and A. Lacaita, "A 2.9-4.0-GHz fractional-N digital PLL with bang-bang phase detector and $560-\mathrm{fs}_{\mathrm{rms}}$ integrated jitter at 4.5-mW power," *IEEE J. Solid-State Circuits*, vol. 46, no. 12, pp. 2745–2758, Dec. 2011.

[99] R. Nonis, W. Grollitsch, T. Santa, D. Cherniak, and N. Da Dalt, "digPLL-Lite: A low-complexity, low-jitter fractional-N digital PLL architecture," *IEEE J. Solid-State Circuits*, vol. 48, no. 12, pp. 3134–3145, Dec. 2013.

[100] A. Elkholy, T. Anand, W.-S. Choi, A. Elshazly, and P. Hanumolu, "A 3.7 mW low-noise wide-bandwidth 4.5 GHz digital fractional-N PLL using time amplifier-based TDC," *IEEE J. Solid-State Circuits*, vol. 50, no. 4, pp. 867–881, April 2015.

[101] R. Staszewski, J. Wallberg, S. Rezeq, C.-M. Hung, O. Eliezer, S. Vemulapalli, C. Fernando, K. Maggio, R. Staszewski, N. Barton, M.-C. Lee, P. Cruise, M. Entezari, K. Muhammad, and D. Leipold, "All-digital PLL and transmitter for mobile phones," *IEEE J. Solid-State Circuits*, vol. 40, no. 12, pp. 2469–2482, Dec. 2005.

[102] E. Temporiti, C. Weltin-Wu, D. Baldi, M. Cusmai, and F. Svelto, "A 3.5 GHz wideband ADPLL with fractional spur suppression through TDC dithering and feedforward compensation," *IEEE J. Solid-State Circuits*, vol. 45, no. 12, pp. 2723–2736, Dec. 2010.

[103] W. Bax and M. Copeland, "A GMSK modulator using a $\Delta\Sigma$ frequency discriminator-based synthesizer," *IEEE J. Solid-State Circuits*, vol. 36, no. 8, pp. 1218–1227, Aug. 2001.

[104] C. Venerus and I. Galton, "A TDC-free mostly-digital FDC-PLL frequency synthesizer with a 2.8-3.5 GHz DCO," *IEEE J. Solid-State Circuits*, vol. 50, no. 2, pp. 450–463, Feb. 2015.

[105] C. Weltin-Wu, G. Zhao, and I. Galton, "A 3.5 GHz digital fractional-N PLL frequency synthesizer based on ring oscillator frequency-to-digital conversion," *IEEE J. Solid-State Circuits*, vol. 50, no. 12, pp. 2988–3002, Dec. 2015.

[106] C. Venerus and I. Galton, "Quantization noise cancellation for FDC-based fractional-N PLLs," *IEEE Trans. Circuits Syst. II*, vol. 62, no. 12, pp. 1119–1123, Dec. 2015.

[107] I. Galton and G. Zimmerman, "Combined RF phase extraction and digitization," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 2, May 1993, pp. 1104–1107.

[108] R. Beards and M. Copeland, "An oversampling delta-sigma frequency discriminator," *IEEE Trans. Circuits Syst. II*, vol. 41, no. 1, pp. 26–32, Jan. 1994.

[109] M. Ferriss and M. Flynn, "A 14 mW fractional-N PLL modulator with a digital phase detector and frequency switching scheme," *IEEE J. Solid-State Circuits*, vol. 43, no. 11, pp. 2464–2471, Nov. 2008.

[110] L. Li, M. Flynn, and M. Ferriss, "A 5.8GHz digital arbitrary phase-setting type II PLL in 65nm CMOS with 2.25° resolution," in *Proc. Asian Solid State Circuits Conference (A-SSCC)*, 2012, pp. 317–320.

[111] M.-J. Park and J. Kim, "Pseudo-linear analysis of bang-bang controlled timing circuits," *IEEE Trans. Circuits Syst. I*, vol. 60, no. 6, pp. 1381–1394, June 2013.

[112] M. Talegaonkar, T. Anand, A. Elkholy, A. Elshazly, R. Nandwana, S. Saxena, B. Young, W. Choi, and P. Hanumolu, "A 4.4-5.4GHz digital fractional-N PLL using $\Delta\Sigma$ frequency-to-digital converter," in *Proc. Symposium on VLSI Circuits (VLSIC)*, June 2014, pp. 1–2.

[113] M. Perrott, M. Trott, and C. Sodini, "A modeling approach for $\Sigma-\Delta$ fractional-N frequency synthesizers allowing straightforward noise analysis," *IEEE J. Solid-State Circuits*, vol. 37, no. 8, pp. 1028–1038, Aug. 2002.

[114] D. Park and S. Cho, "A 14.2 mW 2.55-to-3 GHz cascaded PLL with reference injection and 800 MHz delta-sigma modulator in 0.13 μm CMOS," *IEEE J. Solid-State Circuits*, vol. 47, no. 12, pp. 2989–2998, Dec. 2012.

[115] T.-K. Kao, C.-F. Liang, H.-H. Chiu, and M. Ashburn, "A wideband fractional-N ring PLL with fractional-spur suppression using spectrally shaped segmentation," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2013, pp. 416–417.

[116] D. Schinkel, E. Mensink, E. Klumperink, E. van Tuijl, and B. Nauta, "A double-tail latch-type voltage sense amplifier with 18ps setup+hold time," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2007, pp. 314–315.

[117] R. Nandwana, T. Anand, S. Saxena, S.-J. Kim, M. Talegaonkar, A. Elkholy, W.-S. Choi, A. Elshazly, and P. Hanumolu, "A calibration-free fractional-N ring PLL using hybrid phase/current-mode phase interpolation method," *IEEE J. Solid-State Circuits*, vol. 50, no. 4, pp. 882–895, April 2015.